This electronic thesis or dissertation has been downloaded from the King's Research Portal at https://kclpure.kcl.ac.uk/portal/

Gaussian processes for force fields and wave functions

Glielmo, Aldo

Awarding institution:
King's College London

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without proper acknowledgement.

END USER LICENCE AGREEMENT

Unless another licence is stated on the immediately following page this work is licensed

under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International

licence. https://creativecommons.org/licenses/by-nc-nd/4.0/

You are free to copy, distribute and transmit the work

Under the following conditions:

- Attribution: You must attribute the work in the manner specified by the author (but not in any way that suggests that they endorse you or your use of the work).
- Non Commercial: You may not use this work for commercial purposes.
- No Derivative Works - You may not alter, transform, or build upon this work.

Any of these conditions can be waived if you receive permission from the author. Your fair dealings and other rights are in no way affected by the above.

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 01. Mar. 2021

Gaussian processes for force fields and wave functions

Aldo Glielmo

Department of Physics

King's College London

This dissertation is submitted for the degree of Doctor of Philosophy

January 2019

To my parents, who provided me with the tools that make me a human being,
and to the memory of Sandro, who showed me his way of using them.

Declaration

This dissertation describes work I have carried out between October 2015 and December 2018 at the department of physics of King's College London, under the supervision of Professor Alessandro De Vita (first supervisor from October 2015 to October 2018), Doctor George Booth (first supervisor from October 2015) and Professor Peter Sollich (second supervisor).

This dissertation contains material appearing in the following articles:

- A. Glielmo, C. Zeni, Á. Fekete and A. De Vita. Building nonparametric n-body force fields using Gaussian process regression. Submitted to K. T. Schütt, S. Chmiela, A. von Lilienfeld, A. Tkatchenko, K. Tsuda, and K. R. Müller, editors, *Machine learning for quantum simulations of molecules and materials* (Springer),
- A. Glielmo, C. Zeni and A. De Vita. Efficient nonparametric n-body force fields from machine learning. *Physical Review B*, 97, 2018,
- A. Glielmo, P. Sollich and A. De Vita. Accurate interatomic force fields via machine learning with covariant kernels. *Physical Review B*, 95, 2017.

In addition to the above, I have contributed to the following publications during the course of my PhD:

- M. Cucuringu, P. Davies, A. Glielmo, H. Tyagi. SPONGE: A generalized eigenproblem for clustering signed networks. *Proceedings of Machine Learning Research*, 89, 2019,
- F. Bianchini, A. Glielmo, J. R. Kermode, A. De Vita. Enabling QM-accurate simulation of dislocation motion in γ-Ni and α-Fe using a hybrid multiscale approach. *Physical Review Materials*, 3, 2019,

- N. Gunkelmann, D. Serero, A. Glielmo, M. Montaine, M. Heckel, T. Pöschel. Stochastic nature of particle collisions and its impact on granular material properties. S. Antonyuk, editor, *Particles in contact: Micro Mechanics, Micro Process Dynamics and Particle Collective* (Springer),

- C. Zeni, K. Rossi, A. Glielmo, Á. Fekete, N. Gaston, F. Baletto and A. De Vita. Building machine learning force fields for nanoclusters. *The Journal of Chemical Physics*, 148, 2018,

- M. Heckel, A. Glielmo, N. Gunkelmann, T. Pöschel. Can we obtain the coefficient of restitution from the sound of a bouncing ball?. *Physical Review E*, 93, 2016.

This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and acknowledgements. It has not been submitted in whole or in part for any degree or diploma at this or any other university.

Aldo Glielmo
January 2019

Acknowledgements

Alessandro De Vita is the person that more than any other I here feel the need of thanking. Sandro caught me with his enthusiasm and brilliance when I was eighteen and I had first arrived in London, and he then guided me through my studies with astonishing proximity and dedication.
He taught me what research is, and his particular way of carrying it out made of limitless passion, sustained concentration and creativity, a lot of fun, and a denial of the passage of time. He taught me to think, and not to look for easy answers; to write clearly, and not to hope that people would understand; to do research, and not to publish papers. He made me appreciate the value of what we were doing, and the importance of always doing it in the best possible way. It was impossible not to be in a good mood in his presence, dragged by his uncorrupted thrill towards life. Moments of trouble are however unavoidable for most doctoral students and, in such moments, I knew I could go and knock on his door to find him ready to provide me, unconditionally, with renewed grit and motivation, of which he has been a truly inexhaustible source.
I am deeply grateful to him for all of this, and for the extraordinary amount of time and energy that he has dedicated to my personal and scientific growth. I can only hope that in this endeavour he managed to pass me infinitesimal doses of some of his unique traits.

My research work and scientific growth has greatly benefitted from the interaction with many inspiring researchers, to whom I want to express my gratitude. George Booth welcomed me with enthusiasm when I bashfully investigated possibilities of collaboration. His energy and determination made working with him an extremely rewarding and enjoyable experience. He supported me greatly in my research and, in particular, in the writing of this thesis, of which he also proofread several chapters.
Peter Sollich, offered me precious guidance made of intense meetings during which he could always provide sharp feedback and experienced indications on relevant literature.
James Kermode, also my tutor during by bachelor studies (coinciding with his postdoc), has provided me with conceptual and practical suggestions, and has recently helped me in proofreading several chapters of this thesis.
I furthermore want to thank Francesca Baletto for the closeness and esteem expressed throughout all my years at King's and for the stimulating discussions we had, and I want to acknowledge stimulating discussions also with Anatole von Lilienfeld, Matthias Rupp, Luca Ghiringhelli and Gábor Csányi.

Some people working in the physics department at King's deserve a special thanks for having provided practical help and truthful feedback.
Claudio Zeni has been my closest collaborator. He is, along with me, the most experienced person on the topics covered in this thesis. It hence goes without saying that the background knowledge and the common language we developed in the last years have made scientific discussions held with him always particularly fruitful and often illuminating. He furthermore helped me in practical issues of coding and proofreading.
Ádám Fekete has also closely collaborated with me. The originality of his way of thinking has often stimulated my research and learning efforts towards new directions. He has also been a "computer guru" for me (as well as for the entire department, I believe), able to answer any question and direct me towards the solution of any problem related to computer hardware and software.
Other workers of the seventh floor that I want to mention for scientific support are: Kevin Rossi, Martina Stella, Henry Lambert, Federico Bianchini and Marco Caccin.

My research has received financial support from the Engineering and Physical Sciences Research Council (EPSRC) (Grant No. EP/L015854/1) and the

Office of Naval Research Global (Award No. N62909-15-1-N079), which I gratefully acknowledge, and I thank the center for doctoral training Cross-disciplinary Approaches to Non Equilibrium Systems (CANES), and especially the people behind it, for creating a vibrant and stimulating environment for learning and research.

It is only recently that I started realising what a rare and delicate balance of work efficiency and life enjoyment I have experienced during my years at King's, and I want to try and thank the people who can claim some of the merit for this very fortunate circumstance. I thank Riccardo, a better friend than I had ever hoped to find, for having shared with me everything from life projects to living spaces. I thank Kevin, now an almost ten-year long companion of study and work, for having grown up with me talking about science as well as about philosophy, politics and literature, I thank Silvia for having inspired and enhanced the best part of me, Sara for having been close to me unconditionally and in any moment, Beppe for having faced with me problems seriously while joking and Irene, for having made though days happy. Many others people played an important role in the truly unique "state of grace" I found myself in, I want to thank Davide, Gerard, Michele, Ryan and Stefano, form the first CANES cohort, and Andrea, Alessia, Carla, Celeste and Claudio, who came later but have been equally important.
Finally, I thank my sister Eleonora, for always understanding me deeply, and my parents, Luigi and Susy, for the immense unconditional love they filled me with.

Summary

Gaussian processes for force fields and wave functions

Aldo Glielmo
King's College London

Algorithms capable of extracting information from data are increasingly finding application in condensed matter physics. Two particularly successful application domains have been the automatic construction of atomic force fields and the compact representation of electronic wave functions. In spite of their accuracy, previously proposed data-driven approaches for learning these two quantities often suffer from poor interpretability and transferability.

This thesis develops new accurate and interpretable machine learning models for atomic force fields and for electronic wave functions, based on Gaussian process (GP) regression and on a careful design of GP kernel functions.

To learn atomic force fields, various scalar local energy kernels and matrix-valued force kernels are proposed, all encoding the force field fundamental symmetries (translations, rotations, reflections and permutations) and a controllable degree of complexity provided by the force field interaction order. Tests on a wide range of materials prove the efficiency of the models proposed and show that low order models often represent the best compromise between accuracy and transferability. Furthermore, predictions of low order GP models can be sped up by orders of magnitude, reaching the typical evaluation speed of traditional parametrised potentials.

To learn electronic wave functions, a log-GP model is proposed, along with a set of kernels representing well-defined many-body correlations. Such kernels are benchmarked on the one dimensional Hubbard model with excellent initial results.

Contents

List of Figures

List of symbols and acronyms

Acronyms

DFT	Density functional theory
DFTB	Density functional tight binding
EAM	Embedded atom model
GP	Gaussian process
LJ	Lennard Jones
LOTF	Learn on the fly
MAE	Mean absolute error
MD	Molecular dynamics
MFF	Mapped force field
ML	Machine learning
ML-FF	Machine learning force field
QM	Quantum mechanical
QM/MM	Quantum mechanics/molecular mechanics
VMC	Variational Monte Carlo

Math and Gaussian process conventions

$a \mid \mathbf{a} \mid \mathbf{A} \mid \mathbb{A}$	Scalar, vector, matrix and block matrix
$\mathbf{a}^{\mathrm{T}} \mid \hat{\mathbf{a}}$	The transpose $\mathbf{a}$ and the unit vector in the direction of $\mathbf{a}$
$\sim$	Distributed according to or approximately equal to
$\mathcal{GP}$	Gaussian stochastic process
$\langle\cdot\rangle$	Expectation with respect to a distribution
$\boldsymbol{\theta}$	Vector of hyperparameters
$\ell \mid \sigma_n^2$	Lengthscale and noise hyperparameter
$\mathcal{D}$	Dataset
N	Number of training points
$\mathcal{O}(\cdot)$	Computational complexity

Force field learning

ρ	Local atomic configuration
M	Number of atoms within the cutoff radius of an atomic configuration ρ
$\varepsilon^r \mid \varepsilon(\rho)$	Reference and latent local energy function
$\hat{\varepsilon}(\rho) \mid \hat{\sigma}^2(\rho)$	Mean and variance of the local energy predictive distribution
$k(\rho, \rho')$	Local energy kernel evaluated for configurations ρ and ρ'
$k_n \mid k_n^s \mid k_n^{\neg u}$	Local energy n-body kernel, symmetric n-body kernel and symmetric n-body kernel that is non unique
$k_{MB} \mid k_{MB}^s \mid k_{MB}^{ds}$	Local energy many-body kernels, many-body kernel symmetric over the full orthogonal group and many-body kernel symmetric over a discrete group
$\mathbf{f}^r \mid \mathbf{f}(\rho)$	Reference and latent force on the central atom of a configuration
$\hat{\mathbf{f}}(\rho) \mid \hat{\mathbf{\Sigma}}(\rho)$	Mean and covariance of the force predictive distribution
$\mathbf{K}(\rho, \rho')$	Matrix-valued force kernel evaluated for two configurations
$\mathbf{K}_n^G \mid \mathbf{K}_{MB}^G$	Matrix-valued n-body and many-body force kernels that are covariant over the group G
$\mathcal{P} \mid \mathbf{P}$	Abstract operator and matrix representation of a permutation of atoms of the same chemical species
$\mathcal{R} \mid \mathbf{R}$	Abstract operator and matrix representation of a rotation
$\mathcal{F} \mid \mathbf{F}$	Abstract operator and matrix representation of a reflection
$\mathcal{Q} \mid \mathbf{Q}$	Abstract operator and matrix representation of an element of the orthogonal group (either a rotation or a reflection)
$C_n \mid D_n$	Cyclic and dihedral group of order n

Wave function learning

$\mathbf{x}$	An electronic configuration
$\psi^r \mid \psi(\mathbf{x})$	The reference and the latent wave function
$\psi_S(\mathbf{x})$	Ground state wave function of a quadratic Hamiltonian
$\lambda(\mathbf{x})$	The natural logarithm of the absolute value of the wave function $\psi(\mathbf{x})$ or of the ratio $\psi(\mathbf{x})/\psi_S(\mathbf{x})$
$k(\mathbf{x}, \mathbf{x}')$	Kernel between two electronic configurations $\mathbf{x}$ and $\mathbf{x}'$
$k \mid k_n^d \mid k_c$	Plaquette kernel of size n, distance dependent kernel of size n and complete kernel

Chapter 1

Introduction

DATA, and the fast development of efficient machine learning algorithms capable of exploiting them, are changing the paradigm under which great computational challenges are tackled. Up until ten or fifteen years ago, algorithms were designed with the idea of performing a sequence of instructions, well determined by a programmer, in order to solve a given problem. In more recent times instead, this "paradigm" is shifting towards the design of algorithms that automatically *learn* from data the particular sequence of instructions best suited to solve the problem at hand.

This change can be perhaps best understood by means of an emblematic comparison: that between the software used within the IBM "Deep Blue" computer in 1997 to beat the chess world champion Garri Kasparov [1], and that used twenty years later, in 2016, by Google's "AlphaGo" to beat Lee Sedol, an international champion in the game of Go [2]. While the first algorithm was based on a human-programmed smart search of all possible moves aimed at finding the one yielding to the largest advantage [3–5], within AlphaGo the function yielding to the best move at any given point of the game had been previously learned by the algorithm by both analysing big amounts of previously played matches and by playing against itself [6]. [1]

A similar change is also taking place in the domain of computational condensed matter physics research, where the exploitation of data is recently taking a prominent role. The great computational challenge of this field was well defined by one of the fathers of quantum mechanics, Paul Dirac, who in

[1] Interestingly, a modified version of AlphaGo later repeatedly defeated a modified (and more powerful) version of the Deep Blue chess algorithm [7].

1929 famously wrote [8]:

> The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble. It therefore becomes desirable that approximate practical methods of applying quantum mechanics should be developed, which can lead to an explanation of the main features of complex atomic systems without too much computation.

The *Schrödinger equation*, the fundamental equation governing the behaviour of atoms and molecules, can be considered the most representative of the known laws of physics referred to by Dirac and, as called for by Dirac, much research in computational physics and chemistry has focussed on the design of efficient algorithms to obtain its solution in order to predict physical quantities of interest.

At the risk of oversimplifying it is then possible to carry forward the above comparison and differentiate between two approaches.

The more "traditional" approach has been based on human-designed approximations, giving rise to computer programs capable of simulating larger system for longer time scales. Arguably the most impactful achievement obtained within this category is the development of density functional theory (DFT) [9–11]. DFT, along with the many successful approximations to the unknown exchange and correlation functionals that the theory requires [12], opened up the way to a very vast body of research aimed at predicting or engineering properties of real materials and molecules.

In very recent times, astounding improvements in storing technologies [13, 14] generated a unprecedented availability of data, and decades of growth in processing power made learning algorithms useful in dealing with them. This then opened up the way to a completely new set of approximations and methods to solve the Dirac's problem based on the extraction of information and patterns from large datasets. In the following, some of the most significant developments in this new and quickly growing field are reviewed.

Perhaps the article that first brought the attention of condensed matter physicists on the potential of the exploitation of data was the 2007 Physical Review Letter by Behler and Parrinello [15]. In this work a "machine learning

force field" (ML-FF) was introduced, based on a *neural network* representation of the sought potential energy surface trained directly on a database of DFT calculations. In 2010, Bartók et al. proposed a similar ML-FF model based on *Gaussian processes* rather than on neural networks, which they called "Gaussian approximation potential" [16]. In 2012, *kernel regression* was successfully used by Rupp et al. to learn atomisation energies of molecules and, in the same year, Snyder et al. proposed to use it for learning density functionals [17]. In 2015, Ghiringhelli et al. showed that vast amounts of data combined with a *LASSO regression* could also be used to automatically find the best set of descriptors to characterise materials based on specific properties [18]. Very recently (in 2017), Carleo and Troyer proposed a *reinforcement learning* scheme yielding a very accurate representation of the ground state wave function of a system of electrons, and in the same year Carrasquilla and Melko successfully trained a neural network to recognise quantum phases of matter.

The articles listed above are perhaps the ones that most significantly initiated a research interest in their relative subfields and, as a consequence of the excitement about the initial results achieved, a plethora of algorithms has been subsequently proposed either to improve on the original approaches or to learn other related physical quantities. Notable examples of the latter case are the machine learning algorithms proposed to learn local atomic structures [19], two dimensional atlases of materials [20], free energy surfaces [21] and Green's functions [22].

1.1 Thesis objectives and outline

While the potential of using machine learning (ML) algorithms in physics is clear and unquestioned, as also confirmed by the successful examples listed, there is a common problem arising when using this approach. It is tempting, in fact, to apply ML algorithms in a "black box" fashion and, while this can often still give rise to satisfactory results on specific systems, algorithms developed this way will be crucially limited in their applicability outside of their original, and necessarily narrow, domain of testing. Developing a good understanding of the models that are automatically learned by ML algorithms is, in other words, important for validation and to be able to trust the predictions of these algorithms on previously unseen inputs.

This problem is central to all of the ideas and concepts presented in this thesis, which will attempt to develop the necessary theory for a grounded application of ML to two specific problems in condensed matter physics: that of learning the force field of a system of atoms and that of learning the wave function of a system of electrons.

Force field learning

The main problem considered in this thesis is that of learning an atomic force field i.e., a function giving the forces acting on a system of atoms when provided with the atoms' positions. This is generally done by training on a database containing a set of quantum calculations coming from the solution of the Schröedinger equation, typically solved using a DFT scheme. This is a remarkably difficult task, and the traditional way of carrying it out involves adjusting the parameters of carefully chosen analytic functions in the hope of matching the reference data set [23, 24]. The main difficulty is that developing good parametric models requires a great deal of chemical intuition and patient effort, guided by trial and error steps with no guarantee of success [23–25]. However, for systems and processes in which the approach is fruitful, the development effort is amply rewarded by the opportunity to provide extremely fast and accurate force models [26–29]. The identified functional forms will in these cases contain valuable knowledge on the target system, encoded in a compact formulation that still accurately captures the relevant physics.

Following a different approach, ML-FFs can be constructed. In addition to the two original ML-FFs mentioned above [15, 16], many more schemes have been proposed with the aim of making these algorithms either faster or more accurate [30–33]. In contrast to parametrised force fields, ML-FFs typically do not require a lengthy trial and error to be fitted as they are not constrained to a particular analytic form, thus being much more flexible. However, although ML schemes have been shown to be remarkably accurate interpolators in specific systems, so far they have not become as widespread as it might have been expected. This fact has two main causes. Firstly, ML-FFs can often be based on a "black box" use of ML algorithms, typically involving complex mathematical and algorithmic machinery. As a consequence, they often turn out to be very difficult to interpret and validate, while compact traditional functional forms involve physically descriptive features that make the resulting

model understandable and hence more easily trustworthy. Secondly, standard approaches remain orders of magnitude faster than their ML counterparts [34], and are thus the method of choice when very challenging time or length scales need to be simulated.

This thesis moves in the direction of bridging the gap between the two approaches just compared, with the final aim of producing fast and interpretable models, that are also flexible enough to yield high accuracy, and that can be trained automatically avoiding lengthy tuning by trial and error. A natural framework to tackle the mentioned goal is that of *Bayesian modelling*, as this allows for a transparent inclusion of *prior knowledge* within the algorithms developed, which in turn provides the learning models with greater interpretability and speed. In this thesis, the problem of learning a force field is first formalised in a Bayesian context in Chapter 2: a *prior distribution* over candidates functions is first specified and this is then updated into a *posterior distribution*, which takes the data into account. A Gaussian stochastic process is chosen as the prior distribution, and this gives rise to a particularly flexible regression model called Gaussian process (GP) regression. The quality of a GP regression is predominantly dictated by a proper design of its kernel function, which needs to encode as much prior information as is available on the target function.

Chapters 3 and 4 are devoted to the design and test of a range of kernel functions for energy and force learning equipped with a high degree of prior knowledge. In particular, Chapter 3 develops kernels for learning local energies, encoding properties of smoothness, symmetry and interaction order. Once a local energy function is learned, forces can always be obtained by differentiation, but they can also be learned directly, without passing through an intermediate energy expression.

Chapter 4 develops a general methodology for doing that, and for building matrix-valued kernels that impose the correct rotation and reflection properties on the learned force field. The methodology is then put into practice and a range of kernels is built and tested. The physically based prior information built into the mentioned kernels makes the corresponding models relatively simple to interpret as compared to other state of the art methods, while also maintaining the high accuracy typical of ML-FFs as shown by numerical tests on a variety of materials.

Chapter 5 deals with the problem of selecting a single model, among the many developed either in this thesis or elsewhere, best suited to describe a given system. The results presented in Chapter 5, but also in Chapters 3 and 4, suggest that flexible force fields of low order can often be sufficient to capture the quantum interactions between atoms to a satisfactory accuracy, being more transferable then higher order models.

When low order models are chosen for the description of a system, the predictions coming from a trained GP can be substantially sped up as described in Chapter 6, reaching the speed typical of standard parametrised fitting approaches.

Wave function learning

This thesis also deals with the problem of learning the wave function of a system of electrons. Similarly to the already discussed problem of finding atomic force fields, one can distinguish two approaches to this challenge. The traditional approach involves parameter tuning of specific functional forms capturing important correlations of the sought wave function [35–39], while within more recent approaches based on the use of ML algorithms the wave function is given a generic and very flexible form (notably that of a neural network), automatically adapting to the correlations of the Hamiltonian one wishes to simulate [40, 41].

While the first problem presented (that of learning an atomic force field) is here treated in depth, being the exclusive topic of the Chapters from 3 to 6, the problem of learning a wave function is treated only in Chapter 7. This latter chapter can be read without knowledge of the others, perhaps with the exception of Section 2.2, which explains the fundamentals of GP regression. For learning wave functions a log-GP model is proposed, and a range of GP kernels representing precisely identifiable many-body effects is designed. The accuracy of the proposed method and of the different kernels is tested on data coming from the Hubbard Hamiltonian, a paradigmatic model of strongly interacting electrons.

Chapter 2

Gaussian processes regression

2.1 Introduction

This chapter introduces the necessary background on Gaussian process regression and the way in which it can be successfully applied to build interatomic force fields. Section 2.2 reviews the basic concepts behind standard GP regression, while also introducing the terminology specific to learning local energy functions. Section 2.3 extends the standard framework to its vectorial counterpart, particularly suited to model atomic forces. In contrast to forces and total energies, local energies are not quantum mechanical (QM) observables. However, they still represent a useful concept for constructing GP models as they can be learned from a dataset containing solely forces and/or total energies, and Section 2.4 details how this can be practically done. Finally, Section 2.5 goes through the way in which important physical properties of the local energy function can be included in a GP, focussing on smoothness, symmetries and interaction order.

2.2 Gaussian process regression for scalar quantities

A *local energy* function $\varepsilon(\rho)$ is defined as the energy ε of an atom given a representation ρ of the set of positions of all the atoms surrounding it within a cutoff distance r_c. Such a set of positions is typically called an *atomic environment* or an *atomic configuration*, and ρ could simply be a list of the atomic species and positions expressed in Cartesian coordinates, or any suitably chosen representation of these [30, 32, 42, 43].

Let us assume that a database of reference calculations $\mathcal{D} = \{(\rho_i, \varepsilon_i^r)\}_{i=1}^N$ is available, composed of N local atomic configurations $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_N)^{\mathrm{T}}$ and their corresponding energies $\boldsymbol{\varepsilon}^r = (\varepsilon_1^r, \ldots, \varepsilon_N^r)^{\mathrm{T}}$. A standard assumption, convenient for modelling purposes, is to treat the reference energies $\{\varepsilon_i^r\}$ as the result of the following process

$$\varepsilon_i^r = \varepsilon(\rho_i) + \xi_i, \tag{2.1}$$

where the (latent) true function $\varepsilon(\rho)$ is corrupted by the independent zero-mean Gaussian noise $\xi_i \sim \mathcal{N}(0, \sigma_n^2)$, which can be imagined to model the combined uncertainty associated with both training data and model used. While learning the result of a quantum calculation, the predominant source of uncertainty is typically the *locality error* that results from the assumption of a finite cutoff radius r_c, outside of which atoms are treated as non-interacting[1].

Eq. (2.1) is a common starting point for many fitting approaches. The specificity and power of GP regression over standard parametric approaches lies in the fact that $\varepsilon(\rho)$ is not constrained to any given functional form, but it is rather assumed to be distributed as a Gaussian stochastic process [46], typically with a zero mean:

$$\varepsilon(\rho) \sim \mathcal{GP}(0, k(\rho, \rho')). \tag{2.2}$$

The function k is the *kernel function* of the GP, and it is also called the *covariance function* as it is assumed to provide the correlation

$$k(\rho, \rho') = \langle \varepsilon(\rho)\varepsilon(\rho') \rangle, \tag{2.3}$$

where the brackets here indicate an expectation over the GP distribution. It is important to note here that to be consistent with the above assumption a kernel is required to be a positive semi-definite function. This can be seen from the fact that for any set of real numbers $\{c_i\}$, the defining property of positive semi-definiteness must always be respected as

$$\sum_{ij} c_i k(\rho_i, \rho_j) c_j = \langle (\sum_i c_i \varepsilon(\rho_i))^2 \rangle \geq 0. \tag{2.4}$$

[1]This assumption is necessary in order to define local energy functions, and it typically holds well by virtue of the "nearsightedness" of quantum mechanics [44, 45].

The shorthand notation in Eq. (2.2) signifies that for a vector of input configurations $\boldsymbol{\rho}$, the corresponding vector of local energies $\boldsymbol{\varepsilon} = (\varepsilon(\rho_1), \ldots, \varepsilon(\rho_N))^{\mathrm{T}}$ will be distributed according to a multivariate Gaussian distribution whose covariance matrix is constructed through the given kernel function:

$$\begin{aligned} p(\boldsymbol{\varepsilon} \mid \boldsymbol{\rho}) &= \mathcal{N}(\mathbf{0}, \mathbf{K}) \\ \mathbf{K} &= \begin{pmatrix} k(\rho_1, \rho_1) & \cdots & k(\rho_1, \rho_N) \\ \vdots & \ddots & \vdots \\ k(\rho_N, \rho_1) & \cdots & k(\rho_N, \rho_N) \end{pmatrix}. \end{aligned} \tag{2.5}$$

Since both ξ_i and $\varepsilon(\rho_i)$ are normally distributed, and since the sum of two Gaussian random variables is also Gaussian, one can write down the distribution of the reference energies $\{\varepsilon_i^r\}$ of Eq. (2.1) as a new normal distribution whose covariant matrix is the sum of the original two:

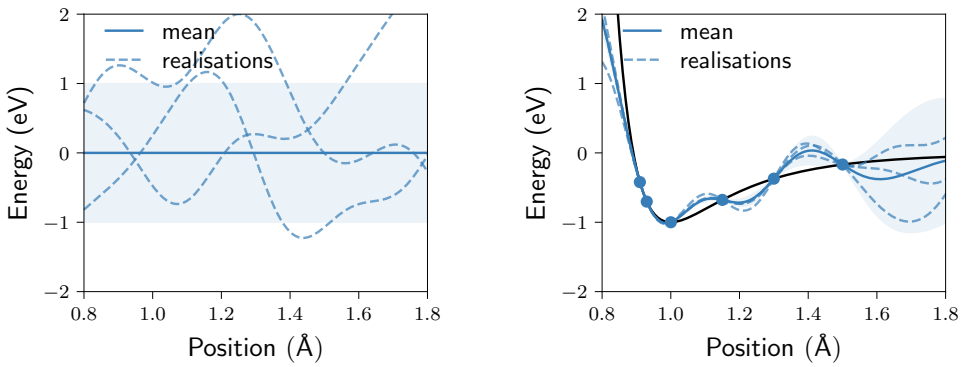
$$\begin{aligned} p(\boldsymbol{\varepsilon}^r \mid \boldsymbol{\rho}) &= \mathcal{N}(\mathbf{0}, \mathbf{C}) \\ \mathbf{C} &= \mathbf{K} + \mathbf{1}\sigma_n^2. \end{aligned} \tag{2.6}$$

Building on this closed form Gaussian expression for the probability of the reference data, it is possible to analytically obtain the *predictive distribution* i.e., the probability distribution of the local energy value ε^* associated with a new target configuration ρ^*, for the given dataset $\mathcal{D} = (\boldsymbol{\rho}, \boldsymbol{\varepsilon}^r)$ (for details on the derivation please refer to Appendix A.1 or Refs. [47, 48]). This is:

$$\begin{aligned} p(\varepsilon^* \mid \rho^*, \boldsymbol{\rho}, \boldsymbol{\varepsilon}^r) &= \mathcal{N}(\hat{\varepsilon}(\rho^*), \hat{\sigma}^2(\rho^*)) \\ \hat{\varepsilon}(\rho^*) &= \mathbf{k}^{\mathrm{T}}(\rho^*)\mathbf{C}^{-1}\boldsymbol{\varepsilon}^r \\ \hat{\sigma}^2(\rho^*) &= k(\rho^*, \rho^*) - \mathbf{k}^{\mathrm{T}}(\rho^*)\mathbf{C}^{-1}\mathbf{k}(\rho^*), \end{aligned} \tag{2.7}$$

where we defined the vector $\mathbf{k}(\rho^*) = (k(\rho^*, \rho_1), \ldots, k(\rho^*, \rho_N))^{\mathrm{T}}$. Notice that the positive semi-definiteness of the kernel function k guarantees the positive definiteness of the matrix $\mathbf{C} = \mathbf{K} + \mathbf{I}\sigma_n^2$, and hence also the existence of its inverse $\mathbf{C}^{-1}$.

The mean function $\hat{\varepsilon}$ of the predictive distribution can be considered a "best guess" for the true underlying function as it minimises the modelled prediction error (cf. Appendix A.2 or Ref. [48]). The mean function is often equivalently written down as a linear combination of kernel functions evaluated



(a) (b)

Figure 2.1: Pictorial view of GP learning of a LJ dimer. Panel (a): mean, standard deviation and random realisations of the prior stochastic process, which represents our belief on the dimer interaction before any data is seen. Panel (b): posterior process, whose mean passes through the training data and whose standard deviation provides a measure of uncertainty.

over all database entries

$$\hat{\varepsilon}(\rho) = \sum_{i=1}^{N} k(\rho, \rho_i)\alpha_i, \tag{2.8}$$

where the coefficients are readily computed as $\alpha_i = (\mathbf{C}^{-1}\boldsymbol{\varepsilon}^r)_i$. The posterior variance of ε^* provides a measure of the uncertainty associated with the prediction, normally expressed as the standard deviation $\hat{\sigma}(\rho)$.

The GP learning process can be thought of as an update of the prior distribution (2.2) into the posterior (2.7). This update is illustrated in Figure 2.1, in which GP regression is used to learn a simple Lennard Jones (LJ) profile from a few data points coming from a dimer.

In particular, Figure 2.1(a) shows the prior GP (Eq. (2.2)) while Figure 2.1(b) shows the *posterior* GP, whose mean and variance functions are those of the predictive distribution in Eq. (2.7). By comparing the two panels one notices that the mean function (equal to zero in the prior process) approximates the true function (black solid line) by passing through the reference calculations. Clearly, the posterior standard deviation (uniform in the prior) shrinks to zero at the points where data is available, to then increase again away from them. Three random function samples are also shown for both prior

and posterior process.

2.3 Gaussian process regression for vector quantities

The force $\mathbf{f}(\rho)$ acting on an atom surrounded by a given environment ρ is a vector quantity. It is hence natural to model it with a *vectorial* GP regression also referred to as "multi-output" or "multi-task" GP regression [49–51]. This rather straightforward extension of the more standard (scalar) method presented in the previous section is outlined in the following.

The starting assumption of the model, analogously to Eq. (2.9), is that for any finite set of configurations $\{\rho_i\}$ the corresponding values of the forces $\{\mathbf{f}(\rho_i)\}$ taken by the vector function $\mathbf{f}(\rho)$ are distributed according to multivariate Gaussian distribution [47]. As a shorthand for this we write:

$$\mathbf{f}(\rho) \sim \mathcal{GP}(\mathbf{0}, \mathbf{K}(\rho, \rho')), \tag{2.9}$$

where $\mathbf{0}$ is the *vector-valued* mean function, here set to zero, and $\mathbf{K}(\rho, \rho')$ is the *matrix-valued* kernel function of the vectorial GP. The kernel $\mathbf{K}(\rho, \rho')$ contains all the information about prior stochastic process as it represents the correlation of the two vectors $\mathbf{f}(\rho)$ and $\mathbf{f}(\rho')$ as a function of the two configurations ρ and ρ':

$$\mathbf{K}(\rho, \rho') = \langle \mathbf{f}(\rho)\mathbf{f}^{\mathrm{T}}(\rho') \rangle, \tag{2.10}$$

where the expectation is taken over the multivariate Gaussian distribution. Similarly to the scalar case (Eq. (2.4)), any kernel $\mathbf{K}$ must be a positive semi-definite matrix-valued function, since for any collection of real valued vectors $\{\mathbf{v}_i\}$ one must have

$$\sum_{ij} \mathbf{v}_i^{\mathrm{T}} \mathbf{K}(\rho_i, \rho_j) \mathbf{v}_j = \langle (\sum_i \mathbf{v}_i^{\mathrm{T}} \mathbf{f}(\rho_i))^2 \rangle \geq 0. \tag{2.11}$$

Once a training database is available, consisting in this case of atomic configurations and reference forces $\mathcal{D} = \{(\rho_i, \mathbf{f}_i^r)\}_{i=1}^N$, and assuming a Gaussian noise process of variance σ_n^2 analogous to Eq. (2.1), the predictive distribution can

again be computed analytically [49]. The result reads:

$$
\begin{aligned}
p(\mathbf{f}^* \mid \rho^*, \boldsymbol{\rho}, \{\mathbf{f}_i^r\}) &= \mathcal{N}(\hat{\mathbf{f}}(\rho^*), \hat{\boldsymbol{\Sigma}}(\rho^*)) \\
\hat{\mathbf{f}}(\rho^*) &= \sum_{ij}^{N} \mathbf{K}(\rho, \rho_i) \mathbb{C}_{ij}^{-1} \mathbf{f}_j^r \\
\hat{\boldsymbol{\Sigma}}(\rho^*) &= \mathbf{K}(\rho^*, \rho^*) - \sum_{ij}^{N} \mathbf{K}^{\mathrm{T}}(\rho^*, \rho_i) \mathbb{C}_{ij}^{-1} \mathbf{K}(\rho^*, \rho_i),
\end{aligned}
\tag{2.12}
$$

where $\mathbb{C} = \mathbb{K} + \mathbb{I}\sigma_n^2$ and blackboard bold characters indicate $N \times N$ block matrices (for instance the *Gram matrix* $\mathbb{K}$ is defined as $(\mathbb{K})_{ij} = \mathbf{K}(\rho_i, \rho_j)$). Similarly, $\mathbb{C}_{ij}^{-1}$ denotes the ij-block of the inverse matrix.

2.4 Local energy from global energies and forces

The forces acting on atoms are well defined local property accessible to QM calculations, easily computed by way of the Hellman-Feynman theorem [52]. As a consequence, the vectorial GP regression framework just described, can in principle be used to learn a force field directly on a database of quantum forces (this will be the topic of Chapter 4). Local atomic energies on the contrary cannot be computed in QM calculations, which can only provide the *total* energy of the full system. However, the material presented in Section 2.2, in addition to being of pedagogical importance, is still useful in practice since local energy functions can be learned from observations of total energies and forces only—this being also the approach used within the well known "Gaussian approximation potential" framework [16, 53].

Mathematically this is possible since any sum, or derivative, of a Gaussian process is also a Gaussian process [47], and the main ingredients needed for learning are hence the covariances (kernels) between these Gaussian variables. In the following, we will see how kernels for total energies and forces can be simply obtained starting from a local energy kernel, and how these can all be used to learn a local energy function.

Kernel functions

Total energy kernels The total energy of a system can be modelled as a sum of the local energies associated to each atomic environment

$$E(\{\rho_a\}) = \sum_{a=1}^{N_a} \varepsilon(\rho_a) \tag{2.13}$$

and if the local energy functions ε in the above equation are distributed according to a zero mean GP, then also the global energy E will be GP variable with zero mean. To calculate the kernel functions $k^{\varepsilon E}$ and k^{EE} providing the covariance between local and global energies and between two global energies one simply needs to take the expectation with respect to the GP of the corresponding products

$$\begin{aligned} k^{\varepsilon E}(\rho_a, \{\rho'_b\}) &= \langle \varepsilon(\rho_a) E(\{\rho'_b\}) \rangle \\ &= \sum_{b=1}^{N'_a} \langle \varepsilon(\rho_a) \varepsilon(\rho'_b) \rangle \\ &= \sum_{b=1}^{N'_a} k(\rho_a, \rho_b). \end{aligned} \tag{2.14}$$

$$\begin{aligned} k^{EE}(\{\rho_a\}, \{\rho'_b\}) &= \langle E(\{\rho_a\}) E(\{\rho'_b\}) \rangle \\ &= \sum_{a=1}^{N_a} \sum_{b=1}^{N'_a} \langle \varepsilon(\rho_a) \varepsilon(\rho'_b) \rangle \\ &= \sum_{a=1}^{N_a} \sum_{b=1}^{N'_a} k(\rho_a, \rho_b). \end{aligned} \tag{2.15}$$

Note that we have allowed the two systems to have a different number of particles N_a and N'_a and that the final covariance functions can be entirely expressed in terms of local energy kernel functions k.

Force kernels The force $\mathbf{f}(\{\rho_a\}^p)$ on an atom p at position $\mathbf{r}_p$ is defined as the derivative

$$\mathbf{f}(\{\rho_a\}^p) = -\frac{\partial E(\{\rho_a\}^p)}{\partial \mathbf{r}_p}, \tag{2.16}$$

where by virtue of the existence of a finite cutoff radius of interaction, only the set of configurations $\{\rho_a\}^p$ that contain atom p within their cutoff function contribute to the force on p.

This quantity is also a GP [47] and the corresponding kernels between forces and between forces and local energies can be easily obtained by differentiation as described in Ref. [47, 54]. They read

$$\mathbf{k}^{\varepsilon\mathbf{f}}(\rho_a, \{\rho_b\}^p) = -\sum_{\{\rho_b\}^q} \frac{\partial k(\rho_a, \rho_b)}{\partial \mathbf{r}_q^{\mathrm{T}}} \quad (2.17)$$

$$\mathbf{K}^{\mathbf{ff}}(\{\rho_a\}^p, \{\rho_b\}^q) = \sum_{\{\rho_a\}^p}\sum_{\{\rho_b\}^q} \frac{\partial^2 k_n(\rho_a, \rho_b)}{\partial \mathbf{r}_p \partial \mathbf{r}_q^{\mathrm{T}}\cdot} \quad (2.18)$$

Total energy-force kernel Learning from both energies and forces simultaneously is also possible. One just needs to calculate the extra kernel $\mathbf{k}^{\mathbf{f}E}$ comparing the two quantities in the database

$$\mathbf{k}^{\mathbf{f}E}(\{\rho_a\}^p, \{\rho'_b\}) = -\sum_{\{\rho_a\}^p}\sum_{b=1}^{N'} \frac{\partial k(\rho_a, \rho_b)}{\partial \mathbf{r}_p}. \quad (2.19)$$

Mixed Gaussian process

To clarify how the kernels described above can be used in practice, it is instructive to look at a simple example. Imagine having a database made up of a single snapshot coming from an *ab initio* molecular dynamics of N atoms, hence containing a single energy calculation and N forces.

Learning using these quantities would involve building a $N+1 \times N+1$ block matrix $\mathbb{K}$ containing the covariance between every pair

$$\mathbb{K} = \begin{pmatrix} k^{EE}(\{\rho_a\}, \{\rho_b\}) & \mathbf{k}^{E\mathbf{f}}(\{\rho_a\}, \{\rho_b\}^1) & \cdots & \mathbf{k}^{E\mathbf{f}}(\{\rho_a\}, \{\rho_b\}^N) \\ \mathbf{k}^{\mathbf{f}E}(\{\rho_a\}^1, \{\rho_b\}) & \mathbf{K}^{\mathbf{ff}}(\{\rho_a\}^1, \{\rho_b\}^1) & \cdots & \mathbf{K}^{\mathbf{ff}}(\{\rho_a\}^1, \{\rho_b\}^N) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{k}^{\mathbf{f}E}(\{\rho_a\}^N, \{\rho_b\}) & \mathbf{K}^{\mathbf{ff}}(\{\rho_a\}^N, \{\rho_b\}^1) & \cdots & \mathbf{K}^{\mathbf{ff}}(\{\rho_a\}^N, \{\rho_b\}^N) \end{pmatrix}. \quad (2.20)$$

As is clear from the above equation, each block is either a scalar (the energy-energy kernel in the top left), a 3×3 matrix (the force-force kernels) or a vector (the energy-force kernels). The full dimension of $\mathbb{K}$ is hence $(3N+1)\times(3N+1)$.

Once such a matrix is built and the inverse $\mathbb{C}^{-1} = [\mathbb{K} + \mathbb{I}\sigma_n^2]^{-1}$ computed, the predictive distribution for the value of the latent local energy variable can be easily written down.

For notational convenience, it is useful to define the vector $\{x_i\}_{i=1}^N$ containing all the quantities in the training database and the vector $\{t_i\}_{i=1}^N$ specifying their type (meaning that t_i is either E or $\mathbf{f}$ depending on the type of data point contained in x_i). With this convention the predictive distribution for the local

energy takes the form

$$\begin{aligned} p(\varepsilon^* \mid \rho, \mathcal{D}) &= \mathcal{N}(\hat{\varepsilon}(\rho), \hat{\sigma}^2(\rho)) \\ \hat{\varepsilon}(\rho) &= \sum_{ij} k^{\varepsilon t_i}(\rho, \rho_i) \mathbb{C}_{ij}^{-1} x_j \\ \hat{\sigma}^2(\rho) &= k(\rho, \rho) - \sum_{ij} k^{\varepsilon t_i}(\rho, \rho_i) \mathbb{C}_{ij}^{-1} k^{t_j \varepsilon}(\rho_j, \rho), \end{aligned} \tag{2.21}$$

where the products between x_j, $\mathbb{C}_{ij}^{-1}$ and $k^{t_j\varepsilon}$ are intended to be between scalars, vectors or matrices depending on the nature of the quantities involved.

2.5 Prior knowledge and kernel functions

Choosing a Gaussian stochastic process as the prior distribution over the energies or forces rather than a parametrised functional form brings a few key advantages. A much sought advantage is that it allows greater flexibility: one can show that in general a GP corresponds to a model with an infinite number of parameters, and with a suitable kernel choice it can act as a "universal approximator"—capable of learning any function if provided with sufficient training data [47]. A second one is a greater ease of design: the kernel function must encode all prior information about the local energy function, but typically contains very few free parameters (called *hyperparameters*) which can be tuned, and such tuning is typically straightforward and can be carried out either by trial and error or via the more principled approaches discussed in Chapter 5. Third, GPs offer a coherent framework to predict the uncertainty associated with the predicted quantities via the posterior variance, absent for classical parametrised force fields.

All this said, the high flexibility associated with GPs can easily become a drawback when examined from the point of view of computational efficiency. Indeed, as it will be clear from the following chapters, for maximal efficiency and out of sample transferability it is important to constrain this flexibility in physically motivated ways, essentially by incorporating prior information in the kernel. In general, this will reduce the dimensionality of the problem e.g., by choosing to learn energy functions of significantly fewer variables than those featuring in the configuration ρ ($3M$ for M atoms in a configuration). To effectively incorporate prior knowledge into the GP kernel it is fundamental to

precisely know the the relation between important properties of the modelled energy and the corresponding kernel properties. These are discussed in the remainder of this chapter, which considers in turn properties of smoothness, invariance to physical symmetries, and interaction order. The focus of the treatment is on the description of scalar kernels for local energies, and the concepts presented will be of fundamental importance in the next chapter, dedicated to the design and use of local energy kernels that are smooth, fully symmetric, and characterised by an adjustable interaction order n. Similar properties apply also to the case of matrix-valued kernels for forces, whose specific design will be the subject of Chapter 4.

Function smoothness

The relation between a given kernel and the smoothness of the random functions described by the corresponding Gaussian stochastic process has been explored in detail [47, 48] and kernels defining functions of arbitrary differentiability have been developed. On one end of the spectrum, the so called *squared exponential* kernel, defines infinitely differentiable functions:

$$k_{SE}(d) = \mathrm{e}^{-d^2/2\ell^2}. \tag{2.22}$$

The letter d here represents the distance between two points of the metric space associated with the function to be learned, e.g. a local energy. At the opposite side of the spectrum, one could use the *absolute exponential* kernel

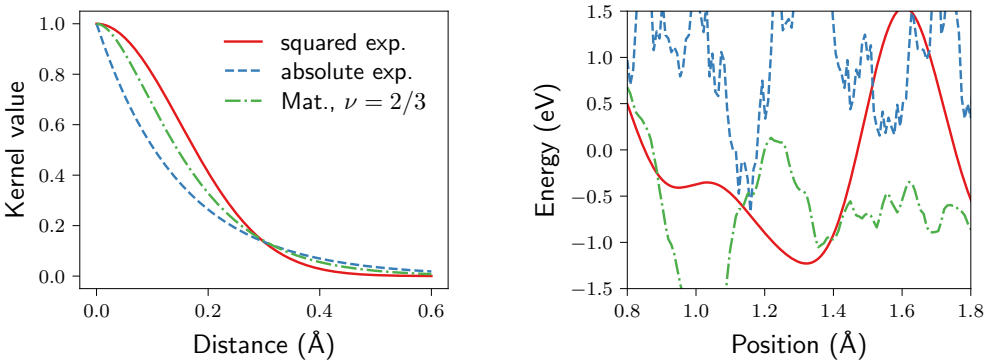
$$k_{AE}(d) = \mathrm{e}^{-d/\ell}, \tag{2.23}$$

defining continuous but not differentiable target functions. Finally, the *Matérn* kernel [47, 48]

$$k_{M,\nu}(d) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{d}{\ell}\right)^{\nu} K_\nu \left(\sqrt{2\nu}\frac{d}{\ell}\right), \tag{2.24}$$

where Γ is the gamma function and K_ν is a modified Bessel function of the second kind, is a generalisation of the other two, and allows a controllable differentiability depending on a parameter ν.

The relation between kernels and the differentiability of the modelled func-



(a) (b)

Figure 2.2: Effect of various kernel functions on the smoothness of the corresponding stochastic processes.

tions is illustrated by Figure 2.2, which shows the three kernels mentioned above (Figure 2.2(a)) along with typical samples from the corresponding GP priors (Figure 2.2(b)).

The absolute exponential kernel has been found useful to learn atomisation energies of molecules [55–57], especially in conjunction with the discontinuous Coulomb matrix descriptor [55]. In the context of modelling useful machine learning force fields, a relatively smooth energy or force function is typically sought. For this reason, the absolute exponential is not appropriate and has never been used while the Matérn covariance has only found limited applicability [58]. In fact, the squared exponential has been almost always preferred, in conjunction with suitable representations ρ of the atomic environment [31, 59–61], and will also be used extensively in this thesis.

Physical symmetries

The following treatment will focus on the invariance properties of local energy kernels. Kernels for forces need to possess analogous properties, with the exception of rotation and reflection symmetry, which will be discussed thoroughly in Chapter 4.

Translations. Physical systems are invariant upon rigid translations of all their components. This basic property is relatively easy to enforce in any

learning algorithm via a local representation of the atomic environments. In particular, it is customary to express a given local atomic environment as the unordered set of M vectors $\{\mathbf{r}_i\}_{i=1}^{M}$ going from the "central" atom to every neighbour lying within a given cutoff radius [32, 42, 43, 59]. It is clear that any representation ρ and any function learned within this space will be invariant upon translations.

Permutations. Atoms of the same chemical species are indistinguishable, and any permutation $\mathcal{P}$ of identical atoms in a configuration necessarily leaves the energy (as well as the force) invariant. Formally, one can write $\varepsilon(\mathcal{P}\rho) = \varepsilon(\rho)\,\forall\mathcal{P}$, and this property corresponds to the kernel invariance

$$k(\mathcal{P}\rho, \mathcal{P}'\rho') = k(\rho, \rho') \quad \forall\mathcal{P}, \mathcal{P}'. \tag{2.25}$$

Typically, the above equality has been enforced either by the use of invariant descriptors [30, 59, 61, 62] or via an explicit invariant summation of the kernel over the permutation group [43, 60, 63], with the latter choice being feasible only when the symmetrisation involves a small number of atoms.

Rotations and reflections. The potential energy associated to a configuration should not change upon any rigid rotation or reflection of the same. By defining $\mathcal{Q}$ to be any element of the orthogonal group (containing both rotations and reflections) this property can be formally written down as $\varepsilon(\mathcal{Q}\rho) = \varepsilon(\rho)\,\forall\mathcal{Q})$. Similarly to permutation symmetry, this invariance is expressed via the following kernel property:

$$k(\mathcal{Q}\rho, \mathcal{Q}'\rho') = k(\rho, \rho') \quad \forall\mathcal{Q}, \mathcal{Q}'. \tag{2.26}$$

The use of invariant descriptors to construct the representation ρ immediately guarantees the above. Typical examples of such descriptors are the symmetry functions originally proposed in the context of neural networks [15, 64], the internal vector matrix [30], or the set of distances between groups of atoms [43, 61, 62]. Alternatively, a "base" kernel k_b can be made invariant with respect to rotations and reflections via the following direct symmetrisation

over the orthogonal grop

$$k(\rho, \rho') = \int_{O(3)} d\mathcal{Q}\, k_b(\rho, \mathcal{Q}\rho'). \tag{2.27}$$

where the integral over all rotations is performed using the normalised measure $d\mathcal{Q}$, invariant over the action of the group. Integrals over continuous symmetry groups as the one just presented are often referred to as "Haar integrals".

The symmetrisation procedure defined in Eq. (2.27) (known as a "transformation integration" in the ML community [65]) was first used to build a potential energy kernel in Ref. [32], and it will be discussed more in depth in the next chapter.

Interaction order

Classical parametrised force fields are sometimes expressed as a truncated series of energy contributions of progressively higher interaction order [23, 24, 28, 29]. The procedure is consistent with the intuition that, as long as the series converges rapidly, truncating the expansion reduces the amount of data necessary for the fitting, and enables a likely higher extrapolation power to unseen regions of configuration space. The lowest truncation order compatible with the target precision threshold is, in general, system dependent, as it will typically depend on the nature of the chemical interatomic bonds within the system. For instance, metallic bonding in a close-packed crystalline system might be described surprisingly well by a pairwise potential, while covalent bonding yielding a zincblende structure can never be, and it will always require three-body interaction terms to be present [43, 59]. Fixing the modelled interaction order can hence be a very powerful way to incorporate prior knowledge into GPs, as also shown in Refs. [43, 59, 61].

The order of a kernel can be defined as the smallest integer n for which the following property holds true:

$$\frac{\partial^n k(\rho, \rho')}{\partial \mathbf{r}_{i_1} \cdots \partial \mathbf{r}_{i_n}} = 0 \qquad \forall\ \mathbf{r}_{i_1} \neq \mathbf{r}_{i_2} \neq \cdots \neq \mathbf{r}_{i_n} \in \rho, \tag{2.28}$$

where $\mathbf{r}_{i_1}, \ldots, \mathbf{r}_{i_n}$ are the positions of any choice of a set of n different surrounding atoms in the configuration ρ. By virtue of linearity, the predicted

local energy in Eq. (2.8) will also satisfy the same property if k does. Thus, Eq. (2.28) implies that the central atom in a local configuration ρ interacts with up to $n-1$ other atoms simultaneously, making the learned energy n-body. For instance, using a 2-body kernel, the force on the central atom due to atom $\mathbf{r}_j$ will not depend on the position of any other atom $\mathbf{r}_{l\neq j}$ belonging to the target configuration $\rho(\{\mathbf{r}_i\})$. Eq. (2.28) can be used directly to check through either numeric or symbolic differentiation if a given kernel is of order n, a fact that might be far from obvious from its analytic form, depending on how the kernel is built.

2.6 Summary

In this chapter, the standard scalar GP regression framework was first reviewed in Section 2.2, while Section 2.3 briefly discussed its vectorial extension. The two approaches were later combined in Section 2.4, which explained how forces and total energies can be used to learn a local energy function.

The importance of a careful design of the kernel—ideally encoding any available prior information on the system studied—was stressed throughout, and Section 2.5 contained an account of the way in which fundamental properties of the target local energy function, such as interaction order, degree of smoothness, as well as invariance over the permutation, translation and orthogonal group, can be included into the kernel function used to model it.

Chapter 3

n-body kernels

3.1 Introduction

The previous chapter introduced the relations between fundamental properties of energies and forces and the corresponding kernel properties. This chapter exploits these relations to build kernels that define smooth and symmetric energy functions of an arbitrary interaction order n. In Section 3.2, these kernels are built through the sequential imposition of properties: from a smooth and permutation invariant representation a range of kernels of finite order is defined and later made rotation and reflection invariant through a Haar integration over the $O(3)$ orthogonal group. Although analytically tractable, the procedure yields kernel functions that are very computationally expensive to evaluate. To improve on this, Section 3.3 follows a different route for the construction of symmetric kernels, which in fact can be also defined directly on invariant degrees of freedom like distances or angles between atoms. This alternative procedure gives rise to more computationally efficient n-body kernels, which however also become unaffordable for high values of n since the cost of summing over all pairs of n-plets grows exponentially with n. Luckily, one can circumvent this exponential wall and increase the order of a symmetric kernel with no computational overhead by raising it to an integer power—obtaining a higher finite oder kernel—or by treating it as the argument of an exponential function—giving rise to a fully many-body kernel.

In this chapter as well as in the following ones it is assumed that atoms are of a single chemical species; the multispecies generalisation of the main n-body kernels proposed is relatively straightforward and is reported in Appendix A.3.

3.2 Building n-body kernels I: O(3) integration

A standard translation and permutation invariant representation of an atomic environment ρ is given by a linear sum of Gaussian functions, each centred on one of the M configuration atoms [32, 42, 59]. Fixing the variance of the Gaussians to be $\ell^2/2$ for later convenience, the mentioned functional representation reads

$$\rho(\mathbf{r}, \{\mathbf{r}_i\}) = \sum_{i=1}^{M} \frac{1}{\ell\sqrt{\pi}} \mathrm{e}^{-\|\mathbf{r}-\mathbf{r}_i\|^2/\ell^2}, \tag{3.1}$$

where $\mathbf{r}$ and $\{\mathbf{r}_i\}_{i=1}^{M}$ are position vectors relative to the central atom of the configuration. This representation guarantees by construction invariance with respect to translations and permutations of atoms.

A 2-body permutation invariant kernel can be obtained as a dot product overlap integral of two configurations [59]:

$$\begin{aligned} k_2(\rho, \rho') &= \int d\mathbf{r}\, \rho(\mathbf{r})\rho'(\mathbf{r}) \\ &= L \sum_{i\in\rho, j\in\rho'} \mathrm{e}^{-\|\mathbf{r}_i-\mathbf{r}'_j\|^2/2\ell^2}, \end{aligned} \tag{3.2}$$

where L is an unessential normalisation factor, omitted for convenience from now on. Interestingly, the above kernel can also be directly defined—without ever passing through the functional representation (3.1)—as the sum of all the squared exponential kernels calculated on the distances between the relative positions in ρ and those in ρ' [66]. The hyperparameter ℓ, which in Eq. (3.1) models the spacial extension of the atoms, can then be observed to also be the lengthscale of a standard squared exponential kernel (Eq. (2.22)) i.e., the typical distance over which the energy function is assumed to presents significant variations.

That the above kernel is a 2-body kernel consistent with the definition of Eq. (2.28) can be checked straightforwardly by explicit differentiation (see Appendix A.4), and its 2-body structure is also deducible from the fact that k_2 is a sum of contributions comparing pairs of atoms in the two configurations: the first pair located at the two ends of vector $\mathbf{r}_i$ in the configuration ρ, and consisting of the central atom and atom i, and the second pair similarly represented by the vector $\mathbf{r}'_j$ in the configuration ρ'.

Higher order n-body kernels can be constructed as finite powers of the

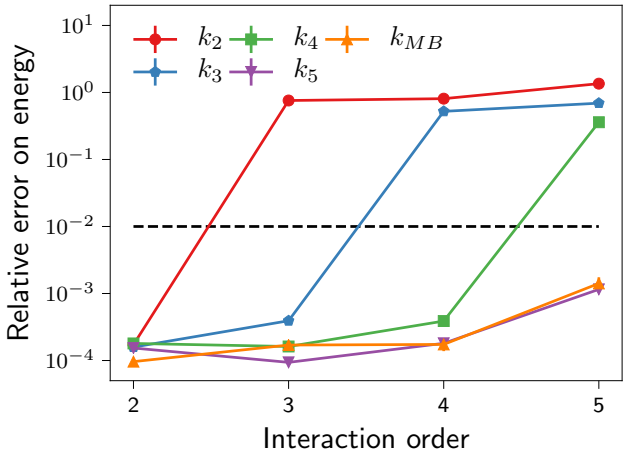


Figure 3.1: GP relative error as a function of the particle's interaction order (2- to 5-body) in a one dimensional system. Learning energies within baseline precision (black dashed line) requires a kernel with an interaction order at least as high as the particles' true interaction order.

2-body base kernel in Eq. (3.2)

$$k_n(\rho, \rho') = k_2(\rho, \rho')^{n-1}, \tag{3.3}$$

where the n-body property (Eq. (2.28)) can once more be easily checked by explicit differentiation (see Appendix A.4). By building n-body kernels using Eq. (3.3), one can avoid the exponential cost of summing over all n-plets that a more naïve kernel implementation would involve. This makes it in principle possible to model any finite interaction order paying only the quadratic computational cost of computing the 2-body kernel in Eq. (3.2). Moreover, one can obtain a fully many-body ("infinite order") kernel by writing a squared exponential kernel on the natural distance $d^2(\rho, \rho') = k_2(\rho, \rho) + k_2(\rho', \rho') - 2k_2(\rho, \rho')$ induced by the "scalar product" k_2. This can be also written down as a formal many-body expansion:

$$\begin{aligned} k_{MB}(\rho, \rho') &= \mathrm{e}^{-d^2(\rho,\rho')/2\ell^2} \\ &= \mathrm{e}^{\frac{-k_2(\rho,\rho)-k_2(\rho',\rho')}{2\ell^2}} \left[1 + \frac{1}{\ell^2}k_2 + \frac{1}{2!\ell^4}k_3 + \frac{1}{3!\ell^6}k_4 + \dots\right]. \end{aligned} \tag{3.4}$$

As all powers of k_2 are contained in the above series, this squared exponential kernel is fully many-body. Interestingly, assuming a smooth underlying

function, the completeness of the series in Eq. (3.4) and the "universal approximator" property of the squared exponential [47, 67] can be immediately seen to imply one another.

To check on these ideas, the proposed kernels are tested in learning the interactions occurring in a simple one dimensional model consisting of n' particles interacting via an *ad hoc* n'-body potential (cf. Appendix A.5 for details on this toy model). We first let the particles interact to generate a configuration database, and then attempt to machine learn these interactions using the kernels just described. Large training databases are used here to test the intrinsic (database-independent) learnability of the interactions. Figure 3.1 illustrates the average relative prediction errors on the local energies of this system incurred by a GP regression based on five different kernels as a function of the interaction order n'. It is clear from the graph that a force field that lets the n' particles interact simultaneously can only be learned accurately with a $(n \geq n')$-body kernel (Eq. (3.3)), or with the many-body squared exponential kernel (Eq. (3.4)) containing all interaction orders.

Rotation and reflection symmetric kernels

To construct n-body kernels useful for applications to real three dimensional systems we need to include rotation and reflection invariance (obtaining the property in Eq. (2.26)). As discussed in Section 2.5 this can be done by performing an integral over the orthogonal group. An invariant or "symmetric" n-body kernel k_n^s can hence be obtained as

$$k_n^s(\rho, \rho') = \int_{O(3)} d\mathcal{Q}\, k_n(\rho, \mathcal{Q}\rho'). \tag{3.5}$$

The use of this type of integral—formally known as a transformation integration in the ML community [65]—was originally proposed in the context of potential energy learning in Ref. [32]. A similar symmetrisation integral can be also envisioned for the many-body base kernel k_{MB} (Eq. (3.4))[43, 59], to define a new many-body kernel k_{MB}^s invariant under all physical symmetries:

$$k_{MB}^s(\rho, \rho') = \int_{O(3)} d\mathcal{Q}\, k_{MB}(\rho, \mathcal{Q}\rho'). \tag{3.6}$$

By virtue of the universal approximation theorem [47, 67] this kernel would be able to learn arbitrary physical interactions with arbitrary accuracy, if provided with sufficient data. Unfortunately, the exponential kernel (3.4) has to date resisted all attempts to carry out the above integration over rotations and reflections analytically, leaving as the only open options numerical integration, or discrete summation over a relevant point group of the system—where the latter can be expected to be particularly efficient if the system presents a clear point symmetry [59]. For a point group G this discrete symmetrisation would take the form

$$k_{MB}^{ds}(\rho, \rho') = \frac{1}{|G|} \sum_{\mathcal{G} \in G} k_{MB}(\rho, \mathcal{G}\rho'). \tag{3.7}$$

On the other hand, the analytic integration of 2- and 3-body kernels has been successfully carried out in different ways. The resulting symmetrized n-body kernel k_n^s will learn faster than its non-symmetrized counterpart k_n, as redundant degrees of freedom have been integrated out. This is because a non-symmetrized n-body kernel k_n must learn functions of $(3n-3)$ variables (translations are taken into account by the local representation based on relative position in Eq. (3.1)). After integration, the new kernel k_n^s defines a smaller and more physically-based space of functions of $(3n-6)$ variables, which is the rotation-invariant functional domain of n interacting particles.

In Ref. [32] the Haar integration of a 3-body over the rotation group $SO(3)$ was carried out using appropriate functional expansions over spherical harmonics. The result was used as an intermediate step for the construction of the widely used "Smooth overlap of atomic positions" (SOAP) kernel [20, 32, 33, 68, 69]. This kernel has a full many-body character, ensured by the prescribed normalisation step [32], which made it possible to use it e.g., to augment to full many-body the descriptive power of a 2- and 3-body explicit kernel expansion [61]. However, the Haar integral over rotations introduced in Ref. [32] as an intermediate kernel construction step could also be seen, if taken on its own, as a transformation integration procedure [65] yielding a symmetrised n-body kernel as defined in Eq. (3.5) above, which would in turn become a higher finite-order kernel if raised to integer powers $\zeta \geq 2$ (see next section).

Carrying out Haar integrals over rotations or reflections is not, in general, an easy task. In the example above, computing a general rotation invariant

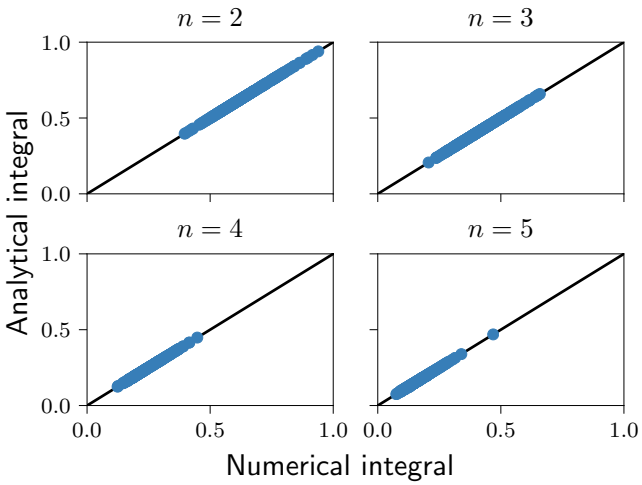


Figure 3.2: Scatter plots showing the values of the integral in (3.9) on a random sample of configurations, computed either by numerical integration or via the analytic expression (Eqs. (3.10, 3.11, 3.13)). Interaction orders from $n = 2$ to $n = 5$ are considered.

n-body kernel via the exact, suitably truncated spherical harmonics expansion procedure of Ref. [32] becomes challenging for $n > 3$. Significant difficulties likewise arise if attempting a "covariant" integration over the orthogonal group, the main subject of the next chapter. In this case, an exact analytic expression was found only for 2- and 3-body matrix-valued kernels [59], with a technique that becomes unviable for $n > 3$. Fortunately, the Haar integration can be avoided altogether, following the simple route of constructing symmetric n-body kernels directly using symmetry-invariant descriptors, as we will see in the next section. The problem of obtaining an analytic Haar integral expression for the general n case remains, however, an interesting one, and is tackled in the remainder of this section following a novel analytic route which fully exploits the Gaussian nature of the configuration expansion in Eq. (3.1).

First, it is useful to express the n-body base kernel of Eq. (3.3) as an explicit product of $(n-1)$ 2-body kernels. The Haar integral in Eq. (3.5) can then be

written as

$$k_n^s(\rho,\rho') = \sum_{\substack{\mathbf{i}=(i_1,\dots,i_{n-1})\in\rho \\ \mathbf{j}=(j_1,\dots,j_{n-1})\in\rho'}} \tilde{k}_{\mathbf{i},\mathbf{j}} \tag{3.8}$$

$$\tilde{k}_{\mathbf{i},\mathbf{j}} = \int_{O(3)} d\mathcal{Q}\, \mathrm{e}^{-\frac{\|\mathbf{r}_{i_1}-\mathbf{Q}\mathbf{r}'_{j_1}\|^2}{2\ell^2}} \dots \mathrm{e}^{-\frac{\|\mathbf{r}_{i_{n-1}}-\mathbf{Q}\mathbf{r}'_{j_{n-1}}\|^2}{2\ell^2}}, \tag{3.9}$$

where now for each of the two configurations ρ, ρ', the sum runs over all n-plets of atoms that include the central atom (whose indices i_0 and j_0 are thus omitted). Expanding the exponents as $(\mathbf{r}_i - \mathbf{Q}\mathbf{r}'_j)^2 = r_i^2 + r_j'^2 - 2\mathrm{Tr}(\mathbf{Q}\mathbf{r}'_j\mathbf{r}_i^{\mathrm{T}})$ allows us to extract from the integral (3.9) a rotation independent constant $\mathcal{C}_{\mathbf{i},\mathbf{j}}$, and to express the rotation-dependent scalar products sum as a trace of a matrix product:

$$\tilde{k}_{\mathbf{i},\mathbf{j}} = \mathcal{C}_{\mathbf{i},\mathbf{j}}\mathcal{I}_{\mathbf{i},\mathbf{j}} \tag{3.10}$$

$$\mathcal{C}_{\mathbf{i},\mathbf{j}} = \mathrm{e}^{-(r_{i_1}^2 + r_{j_1}'^2 + \dots r_{i_{n-1}}^2 + r_{j_{n-1}}'^2)/2\ell^2} \tag{3.11}$$

$$\mathcal{I}_{\mathbf{i},\mathbf{j}} = \int_{O(3)} d\mathcal{Q}\, \mathrm{e}^{\mathrm{Tr}(\mathbf{Q}\mathbf{M}_{\mathbf{i},\mathbf{j}})}, \tag{3.12}$$

where the matrix $\mathbf{M}_{\mathbf{i},\mathbf{j}}$ is the sum of the outer products of the ordered vector couples in the two configurations: $\mathbf{M}_{\mathbf{i},\mathbf{j}} = (\mathbf{r}'_{j_1}\mathbf{r}_{i_1}^{\mathrm{T}} + \dots + \mathbf{r}'_{j_{n-1}}\mathbf{r}_{i_{n-1}}^{\mathrm{T}})/\ell^2$. The integral (3.12) occurs in the context of multivariate statistics as the generating function of the non-central Wishart distribution [70]. As shown in [71], it can be expressed as a power series in the symmetric polynomials $\alpha_1 = \sum_i^3 \mu_i$, $\alpha_2 = \sum_{i<j}^3 \mu_i\mu_j$ and $\alpha_3 = \mu_1\mu_2\mu_3$ of the eigenvalues $\{\mu_i\}_{i=1}^3$ of the symmetric matrix $\mathbf{M}_{\mathbf{i},\mathbf{j}}^{\mathrm{T}}\mathbf{M}_{\mathbf{i},\mathbf{j}}$:

$$\begin{aligned} \mathcal{I}_{\mathbf{i},\mathbf{j}} &= \sum_{p_1,p_2,p_3} A_{p_1p_2p_3}\alpha_1^{p_1}\alpha_2^{p_2}\alpha_3^{p_3} \\ A_{p_1p_2p_3} &= \frac{\pi\, 2^{-(1+2p_1+4p_2+6p_3)}(p_1+2p_2+4p_3)!}{p_1!p_2!p_3!\Gamma(\frac{3}{2}+p_1+2p_2+3p_3)\Gamma(1+p_2+2p_3)} \\ &\quad \times \frac{1}{\Gamma(\frac{1}{2}+p_3)(p_1+2p_2+3p_3)!}. \end{aligned} \tag{3.13}$$

Remarkably, in this result (whose exactness is checked numerically in Figure 3.2) the integral over the orthogonal group does not depend on the order n of the base kernel, once the matrix $\mathbf{M}_{\mathbf{i},\mathbf{j}}$ is computed. This is not the case

for previous approaches to integrating over rotations [32, 59] that need to be reformulated with increasing and eventually prohibitive difficulty each time the order n needs to be increased. However, the final expression given by Eqs. (3.10-3.13) is still a relatively complex and computationally expensive function of the atomic positions. Fortunately such complexity can be largely avoided altogether if equally accurate kernels can be built by physical intuition at least for the most relevant lowest n orders, as discussed in the next section.

3.3 Building n-body kernels II: n-body feature spaces

The practical effect of the Haar integration (3.5) is the elimination of the three spurious rotational degrees of freedom. The same result can often be achieved by selecting a group of symmetry-invariant degrees of freedom for the system, typically including the distances and/or bond angles found in local atomic environments, or simple functions of these. Appropriate symmetric kernels can then simply be obtained by defining a similarity measure *directly* on these invariant quantities [30, 55, 61, 72]. To construct symmetry invariant n-body kernels with $n = 2$ and $n = 3$ we can choose these degrees of freedom to be just interparticle distances:

$$k_2^s(\rho, \rho') = \sum_{\substack{i\in\rho \\ j\in\rho'}} \mathrm{e}^{-(r_i - r'_j)^2/2\ell^2}, \tag{3.14}$$

$$k_3^s(\rho, \rho') = \sum_{\substack{i_1>i_2\in\rho \\ j_1>j_2\in\rho'}} \sum_{\mathbf{P}\in\mathcal{P}} \mathrm{e}^{-\|(r_{i_1}, r_{i_2}, r_{i_1 i_2})^{\mathrm{T}} - \mathbf{P}(r'_{j_1}, r'_{j_2}, r'_{j_1 j_2})^{\mathrm{T}}\|^2/2\ell^2}, \tag{3.15}$$

where r_i indicates the Euclidean norm of the relative position vector $\mathbf{r}_i$, and the sum over all permutations of three elements $\mathcal{P}$ ($|\mathcal{P}| = 6$) ensures the permutation invariance of the kernel (as defined in Eq. (2.25)).

Since these kernels learn functions of low-dimensional spaces, their exact analytic form is not essential for performance, as many well behaved functions of the relevant degrees of freedom are likely to give equivalent converged results in the rapidly reached large-database limit. This equivalence can be neatly observed in Figure 3.3, which reports the performance of 2- and 3-body kernels built either directly over the set of distances (Eqs. (3.14) and (3.15)) or via the exact Haar integral (Eqs. (3.8-3.13)). As the test system is crystalline

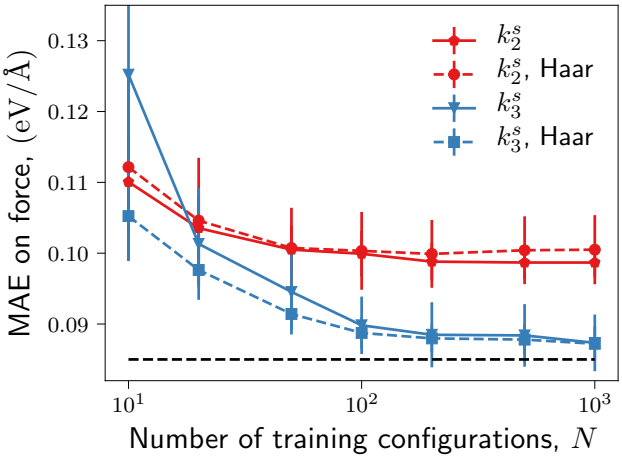


Figure 3.3: Learning curves for 2- and 3-body kernels obtained either via a Haar integration (Eqs. (3.8-3.13)), or directly specifying a similarity kernel function on the effective degrees of freedom (Eqs. (3.14, 3.15)). The mean absolute error (MAE) is calculated as the mean of the absolute value of the vector difference between predicted and reference force.

silicon, 3-body kernels are better performing. However, since convergence of the 2- and 3-body feature space is quickly achieved (at about $N = 50$ and $N = 100$ respectively), there is no significant performance difference between $O(3)$-integrated n-body kernels and physically motivated ones. Consequently, for low interaction orders, simple and computationally fast kernels like the ones in Eqs. (3.14, 3.15) are always preferable to more complex (and heavier) alternatives obtained via Haar integration (e.g., the one defined by Eqs. (3.8-3.13) or those found in Refs. [32, 59]).

An immediate generalisation of Eqs. (3.14,3.15) is given by the construction of an arbitrary symmetric n-body kernel as

$$k_n^s(\rho, \rho') = \sum_{\substack{i_1>\cdots>i_{n-1}\in\rho \\ j_1>\cdots>j_{n-1}\in\rho'}} \tilde{k}_n(\mathbf{q}_{i_1,\ldots,i_{n-1}}, \mathbf{q}'_{j_1,\ldots,j_{n-1}}), \tag{3.16}$$

where the components of the feature vectors $\mathbf{q}$ are the chosen symmetry-invariant degrees of freedom describing the n-plets of atoms. The $\mathbf{q}$ feature vectors are required to be $(3n-6)$ dimensional for all n, except for $n = 2$, where they become scalars. In practice, for $n > 3$ selecting a suitable set of invariant degrees of freedom is not trivial. For instance, for $n = 4$ the set of

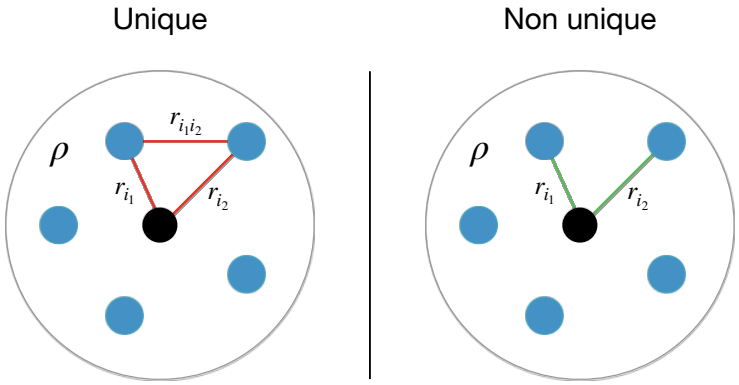


Figure 3.4: Unique interaction (left panel) associated with the 3-body kernel k_3^s in Eq. (3.15) compared with the non-unique 3-body interaction (right panel) associated with the kernel $k_3^{\neg u} = (k_2^s)^2$ in Eq. (3.17), which is a function of two distances only (see text).

six unordered distances between four particles does not specify their relative positions unambiguously, while for $n > 4$ the number of distances associated with n atoms exceeds the target feature space dimension $(3n-6)$. Meanwhile, the computational cost of evaluating the full sum in Eq. (3.16) very quickly becomes prohibitively large as the number of elements in the sum grows exponentially with n.

Consequently, building high order kernels using Eq. (3.16) is only practical when there are very few atoms in atomic configurations (low M). However, the order of an *already symmetric* n-body kernel can be augmented with no computational overhead by generating a derived kernel through simple exponentiation to an integer power, although this is achieved at the cost of losing the *uniqueness* [32, 73, 74] of the representation. This can be easily understood by means of an example, graphically illustrated in Figure 3.4. Let us consider the 2-body symmetric kernel k_2^s (Eq. (3.14)) which learns a function of just a single distance, and therefore treats the r_i distances between the central atom and its neighbors independently. Its square has the form

$$k_3^{\neg u}(\rho, \rho') = \sum_{\substack{i_1 > i_2 \in \rho \\ j_1 > j_2 \in \rho'}} \mathrm{e}^{-\frac{(r_{i_1} - r'_{j_1})^2}{2\ell^2}} \mathrm{e}^{-\frac{(r_{i_2} - r'_{j_2})^2}{2\ell^2}}. \tag{3.17}$$

Such a kernel will be able to learn functions of two distances r_{i_1}, r_{i_2} from the

central atom of the target configuration ρ (see Figure 3.4) and thus will be a 3-body kernel in the sense of Eq. (2.28). However, it will not be able to resolve angular information, as rotating the atoms in ρ around the origin by independent, arbitrary angles will yield identical predictions.

Extending this line of reasoning, it is easy to show that squaring a symmetric 3-body kernel yields a kernel that can capture interactions up to 5-body, although again non-uniquely. This has often been done in practice by squaring the SOAP integral [61, 69]. Raising a 3-body "input" kernel to an arbitrary integer power $\zeta \geq 2$ yields an n-body output kernel of order $2\zeta + 1$:

$$k^{\neg u}_{n=2\zeta+1}(\rho, \rho') = k^{s}_{3}(\rho, \rho')^{\zeta}. \tag{3.18}$$

This kernel is also non-unique as it will learn a function of only 3ζ variables, while the total number of relevant n-body degrees of freedom ($3n-6 = 6\zeta-3$) is always larger than this. Substituting 3 with any n' order of the symmetrized input kernel will similarly generate a $k^{\neg u}_{n=(n'-1)\zeta+1} = k^{s}_{n'}(\rho, \rho')^{\zeta}$ kernel of order $n = (n'-1)\zeta + 1$. A simple calculation reveals that, also in the general case, the number of variables on which $k^{\neg u}_{n}$ is implicitly built is $(3n'-6)\zeta$, always smaller than the full dimension of n-body feature space $(3n'-3)\zeta - 3$ (as expected, the two become equal only for the trivial exponent $\zeta = 1$).

In practice, the non-uniqueness issue appears to be a severe problem only when the input kernel is a 2-body kernel, and as such it depends only on the radial distances from the central atoms occurring in the two atomic configurations (cf. Eq. (3.15) and Figure 3.4). In this case the non unique output n-body kernels will depend on ζ-plets of radial distances, and will miss angular correlations encoded in the training data [43]. On the contrary, a symmetric 3-body kernel (Eq. (3.15)) contains angular information on all triplets in a configuration and, using this kernel as input, the output kernel can be able to capture higher interaction orders (as confirmed e.g., by the numerical tests performed in Ref. [32]).

None of the kernels obtained as finite powers of some symmetric lower-order kernels is a many-body one (they will all satisfy Eq. (2.28) for some finite n). However, an attractive immediate generalisation consists of substituting any squaring or cubing with full exponentiation similarly to what was done in Eq. (3.4).

Indeed, one could build a symmetric many-body kernel as a squared exponential on the 3-body invariant distance $d_s^2(\rho,\rho') = k_3^s(\rho,\rho) + k_3^s(\rho',\rho') - 2k_3^s(\rho,\rho')$, obtaining

$$k_{MB}^s(\rho,\rho') = \mathrm{e}^{-(k_3^s(\rho,\rho)+k_3^s(\rho',\rho')-2k_3^s(\rho,\rho'))/2\ell^2}. \tag{3.19}$$

It is clear from the infinite expansion in Eq. (3.4) that this kernel is a many-body one in the sense of Eq. (2.28), and is also fully symmetric. As is also the case for all finite-power kernels, the computational cost of this many-body kernel will depend on the order n' of the input kernel (3 in the present example) as the sum in Eq. (3.16) only runs on the atomic n'-plets (here, triplets) in ρ and ρ'. This new kernel is not *a priori* known to neglect any order of interaction that might occur in a physical system and thus be encoded in a reference QM training database.

Another way to inexpensively obtain a many-body kernel is by normalisation of an explicit finite order one

$$k_{MB}^s(\rho,\rho') = \frac{k_3^s(\rho,\rho')}{\sqrt{k_3^s(\rho,\rho)k_3^s(\rho',\rho')}}. \tag{3.20}$$

The denominator makes this many-body in the sense of Eq. (2.28) (as is also the case for the SOAP kernel, while no Haar integration is needed here). Incidentally, the above can also be seen to be a squared exponential kernel on the distance induced by the scalar product kernel given by the natural logarithm of $k_3^s(\rho,\rho')$.

This section provided a definition for an n-body kernel, and proposed a general formalism for building n-body kernels by exact Haar integration over the orthogonal group. A class of simpler kernels based on invariant features was defined, also n-body according to the previous definition. As both approaches become computationally expensive for high values of n, it was pointed out that n-body kernels can be built as powers of lower-order input n'-body kernels, with no additional computational overhead. While such a procedure comes at the cost of sacrificing the uniqueness property of the descriptor, it also suggests how to build, by full exponentiation, a many-body symmetric kernel. For many applications, however, using a finite-order kernel will provide the best option, as suggested by the numerical tests reported in the next section.

kernel	order	name	relevant eq.
k_2^s	2	2-body	3.14
$k_3^{\neg u}$	3	3-body, non-unique	3.17
k_3^s	3	3-body	3.15
$k_5^{\neg u}$	5	5-body, non-unique	3.18
k_{MB}^{ds}	∞	many-body, discrete symm.	3.7

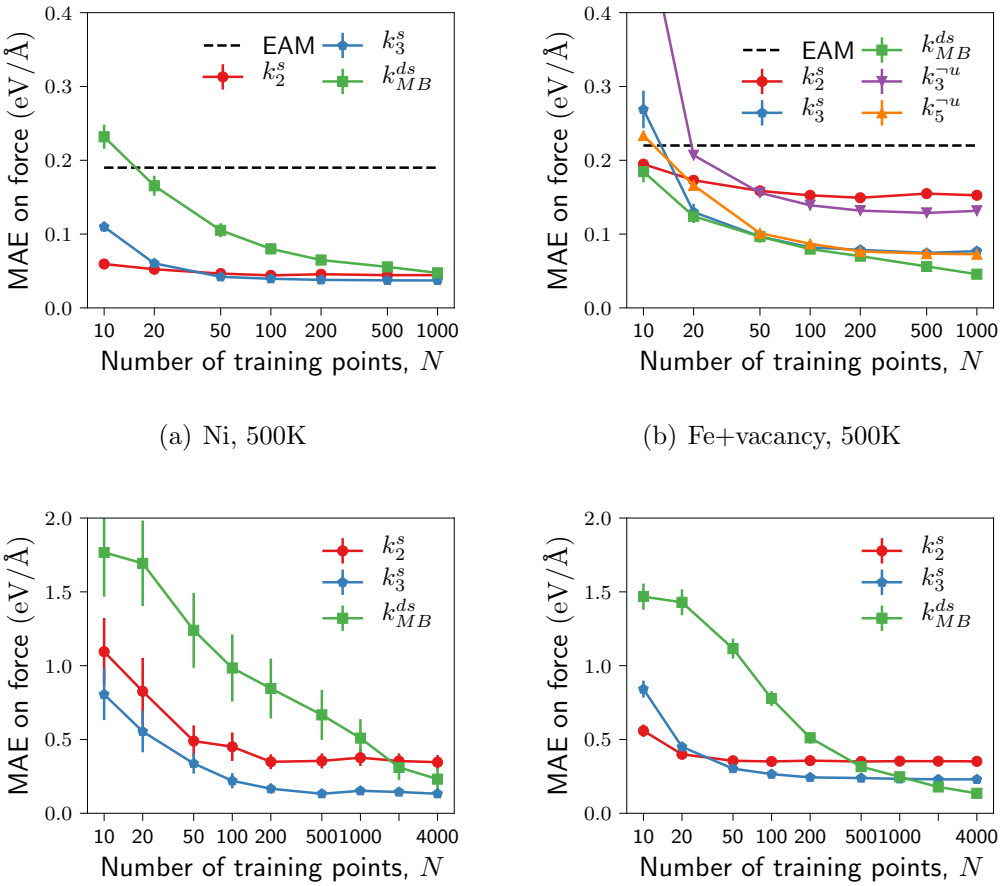
Table 3.1: Some of the kernels proposed. The kernel k_{MB}^{ds} was obtained by a discrete sum over the O_{48} crystalline group.

3.4 Tests on real systems

The performance of some of the kernels proposed in the last section is here tested on a range of realistic materials described at the DFT level of accuracy (please cf. Appendix A.6 for more details on the datasets used).

In this section, as well as in Figure 3.3 and in the rest of this thesis, the Euclidean norm of the vector difference $\|\mathbf{f}_i^r - \hat{\mathbf{f}}(\rho_i)\|$ between reference force $\mathbf{f}_i^r$ and predicted force $\hat{\mathbf{f}}(\rho_i)$ is used as a measure or error. The mean value of this quantity across a randomly sampled "test" dataset (not containing any training point) is defined as the mean absolute error (MAE) on force. Calculating the MAE for different randomly sampled training and test sets provides the standard deviation and hence the errors bars plotted.

The kernels considered are listed for convenience in Table 3.1, while their performance is compared in Figure 3.5. The figure reveals some general trends. 2-body kernels can be trained very quickly, as good convergence can be attained already with $N \sim 100$ training configurations. The 2-body representation is a very good descriptor for a few materials under specific conditions, while its overall accuracy is ultimately limited. This will yield e.g., excellent force accuracy for a close-packed bulk system like crystalline Nickel (panel (a)), and reasonable accuracy for a defected α-Fe system (panel (b))—whose bcc structure is however metastable if just pair potentials are used. Accuracy improves dramatically once angular information is acquired by training 3-body kernels. These can accurately describe forces acting on iron atoms in the bulk α-Fe system containing a vacancy (panel (b)) and those acting on carbon atoms in both diamond and graphite (panel (c)). However, 3-body GPs need larger training databases. Also, atoms participate in many more triplets than



(a) Ni, 500K

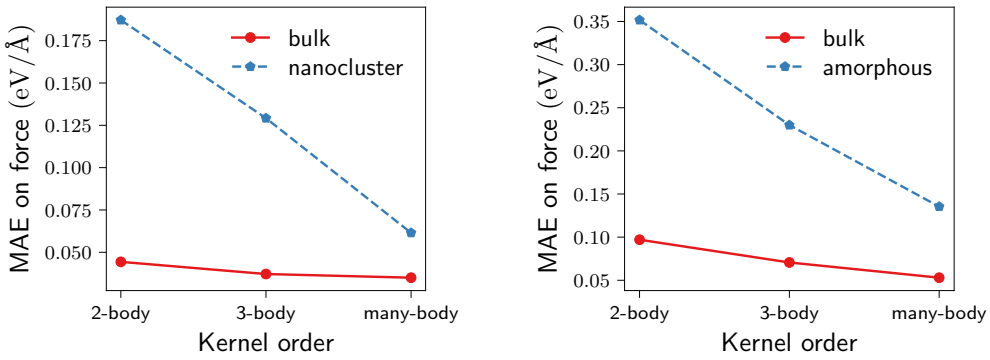
(b) Fe+vacancy, 500K

(c) C

(d) a-Si, 650K

Figure 3.5: Learning curves reporting the MAE on force as a function of the training set size, for different materials and kernels of increasing order. The systems considered are: (a) crystalline nickel, 500K; (b) iron with a vacancy, 500K; (c) diamond and graphite, mixed temperatures and pressures; and (d) amorphous silicon, 650K. The embedded atom model (EAM) potentials in panels (a) and (b) (proposed in Refs. [26] and [75] respectively) were calibrated to match the lattice constants of the reference DFT data. For extra details on the datasets used cf. Appendix A.6

simple bonds in their standard environments contained in the database, which will make 3-body kernels slower than 2-body ones for making predictions by GP regression. Both problems would extend, getting worse, to higher values of n, as summing over all database configurations and all feature n-plets in each database configuration will make GP predictions progressively slower. However, complex materials where high-order interactions presumably play



(a) (b)

Figure 3.6: Converged error as a function of the kernel's order for Ni and Si systems. In panel (a) crystalline nickel is compared to a nickel nanocluster while in panel (b) crystalline silicon is compared to amorphous silicon.

a significant role should be expected to be well described by GPs based on a many-body kernel. This is verified here in the case of amorphous Silicon (panel (d)).

Figure 3.5 (b) also shows the performance of some non-unique kernels. As discussed above, these are options to increase the order of an input kernel avoiding the need to sum over the correspondingly higher order n-plets. Our tests indicate that the GPs generated by non-unique kernels sometimes improve appreciably on the input kernels' performance: e.g., the error incurred by the 2-body kernel of Eq. (3.14) in the Fe-vacancy system is higher than that associated with its square, the non-unique 3-body kernel of Eq. (3.17). Unfortunately, but not surprisingly, the improvement can be in other cases modest or nearly absent, as exemplified by comparing the errors associated with the 3-body kernel and its square (the non-unique 5-body kernel) in the same system.

Overall, the analysis of Figure 3.5 suggests that one way to select an optimal kernel is by comparing the learning curves of the various n-body kernels and the many-body kernel over the available QM database: the comparison will reveal the simplest (most informative, lowest n) description that is still compatible with the error level deemed acceptable in the simulation. Identifying the n value best suited for the description of a given material system can also be done in practice by monitoring how the converged error varies as a function of

the kernel order. Plots illustrating this behaviour are provided in Figure 3.6, where nickel and silicon systems are considered respectively in panels (a) and (b). In each plot the more complex system (a Ni cluster and an amorphous Si system, respectively) display a high accuracy gain (larger negative slope) when the kernel order is increased, while the relatively simpler cristalline Ni and Si systems show a practically constant trend on the same scale. The issue of choosing the kernel order n can also be tackled in the context of Bayesian theory and it will be explored in greater depth in Chapter 5.

Trading transferability for accuracy by training the kernels on a QM database appropriately tailored for the target system (e.g., restricted to just bulk or simply-defected system configurations sampled at the relevant temperatures as done in the Ni and Fe-systems of Figure 3.5) will enable surprisingly good accuracy even for low n values. This should be expected to systematically improve on the accuracy performance of classical potentials involving nonlinear parameter fitting, as exemplified by comparing the errors associated with n-body kernel models and the average errors of state-of-the-art embedded atom model (EAM) potential [26, 75] (see e.g., panels (a) and (b) of Figure 3.5).

3.5 Summary

This chapter provided explicit mathematical expressions for a set of kernels encoding smoothness and invariance properties desirable to any molecular dynamics (MD) force field and an adjustable parameter n controlling the modelled interaction order. Section 3.2 presented a systematic way to construct such kernels by enforcing relevant properties one after the other. Permutation invariance is encoded into a functional representation of atomic environments based on sums of Gaussian functions, a base n-body kernel is then defined starting from an overlap integral of the product of two environments, and invariance over rotations and reflections is then imposed via a Haar integration over the orthogonal group. All the steps can be performed analytically but the process still results in very computationally expensive functions of the atomic positions. To improve on this problem, Section 3.3 covered an alternative way to build kernels with equivalent features. In this parallel approach, n-body kernels are built directly on the invariant degrees of freedom of groups of n atoms. This results in simple, and computationally more efficient kernels, whose per-

formance is shown to be equivalent to that achieved by the Haar-integrated kernels.

Although this alternative procedure also becomes computationally demanding for $n > 3$, higher order symmetric kernels can be obtained by exponentiating lower order ones to integer powers. Kernels obtained in this way will be "non-unique" i.e., incapable of learning an arbitrary physical interaction of the same order. However, this issue can be imagined to be a severe limitation to the final accuracy only when the input kernel is 2-body and all the angular information is absent.

Tests on a range of DFT materials proved the effectiveness of the proposed kernels, all improving on state of the art parametric potentials in terms of error on the target forces. The relative accuracy of the n-body kernels was also found to be highly dependent on the material under consideration, suggesting that the best n will always be system dependent

Chapter 4

Covariant kernels

4.1 Introduction

The concept of local energy extensively used in the last chapter, is not necessary for the construction of accurate machine learning force fields. In fact, in learn on the fly (LOTF) [76] molecular dynamics applications, the high-accuracy target and local interpolation character of force predictions makes it appealing to learn forces directly rather than learning a local energy scalar field first and then deriving forces by differentiation. One way to accomplish this is by using GP regression to separately learn individual force components [30, 31, 77, 78]. This approach might result in computationally fast algorithms but is intrinsically limited by the impossibility of exploiting the strong correlation existing between the three components of a force vector, typically induced by the equivalence of forces related by a rigid rotation or reflection of the configuration space. In order to exploit this basic symmetry, it is once again necessary to impose this constraint into the (now matrix-valued) kernel function.

This chapter deals with the definition, construction and use of "covariant kernels" i.e., kernels that incorporate the covariant behaviour of the function to be learned. In addition to being surely of potential use in LOFT simulations, covariant kernels built on the ideas introduced in this chapter have also found great applicability in efficiently learning other physical vectors (or higher order tensors) that are also covariant under a given symmetry transformation [79, 80].

Section 4.2 introduces the definition of a covariant kernel, and proves that

a kernel possessing the covariance property learns vector valued functions with the correct behaviour under the given symmetry operation. Section 4.3 proposes a general procedure for building covariant kernels from non covariant ones, based on a Haar integration over the orthogonal group similar to the one thoroughly discussed in Chapters 2 and 3. In Section 4.4 this procedure is put into practice to build n-body and many-body covariant kernels for learning forces in one, two and three dimensional spaces. While the lower dimensional examples serve mainly to provide illustrative scenarios for understanding the effect of covariance imposition, the three dimensional covariant kernels obtained can be used to accurately learn forces in realistic materials. This is done in Section 4.5, where the performance of covariant kernels is tested in nickel, iron and silicon systems.

4.2 Kernel covariance

The symbol $\mathcal{Q}$ here represents a member of the orthogonal group. In particular, $\mathcal{Q}$ can be either rotation (for which the symbol $\mathcal{R}$ will be used) or a reflection (represented by the symbol $\mathcal{F}$) acting on an atomistic configuration ρ [1]. There are two properties that the trained GP should respect once configurations are transformed by an operator $\mathcal{Q}$ (represented by a matrix $\mathbf{Q}$).

Property 1 If the target configuration ρ is transformed to $\mathcal{Q}\rho$, the GP predicted mean and covariance must transform accordingly:

$$\begin{aligned} \hat{\mathbf{f}}(\mathcal{Q}\rho) &= \mathbf{Q}\hat{\mathbf{f}}(\rho) \\ \hat{\mathbf{\Sigma}}(\mathcal{Q}\rho) &= \mathbf{Q}\hat{\mathbf{\Sigma}}(\rho)\mathbf{Q}^{\mathrm{T}}. \end{aligned} \tag{4.1}$$

Property 2 The predicted mean and covariance must not change if we arbitrarily transform the configurations in the database ($\mathcal{D} \to \tilde{\mathcal{D}} = \{(\mathcal{Q}_i\rho_i, \mathbf{Q}_i\mathbf{f}_i^r)\}$) with any chosen set of roto-reflections $\{\mathcal{Q}_i\}$.

A special class of kernel functions that automatically guarantees these two properties can be now introduced: a kernel is defined to be *covariant* over a

[1] Although rotations and reflections are the focus of the present chapter, the general theoretical results on kernel covariance developed in this section apply to the general case of arbitrary transformations.

given symmetry group (here the orthogonal group) if it transforms as follows under the action of two arbitrary elements $\mathcal{Q}$ and $\mathcal{Q}'$ of that group:

$$\mathbf{K}(\mathcal{Q}\rho, \mathcal{Q}'\rho') = \mathbf{Q}\mathbf{K}(\rho, \rho')\mathbf{Q}'^{\mathrm{T}}. \tag{4.2}$$

That a covariant kernel imposes Property 1 follows directly from the defining equations for posterior mean and variance. For instance, for the predicted force on a rotated configuration $\mathcal{Q}\rho$ one simply obtains

$$\begin{aligned} \hat{\mathbf{f}}(\mathcal{Q}\rho) &= \sum_{ij}^{N} \mathbf{K}(\mathcal{Q}\rho, \rho_i)[\mathbb{K} + \mathbb{I}\sigma_n^2]^{-1}_{ij}\mathbf{f}_j^r \\ &= \sum_{ij}^{N} \mathbf{Q}\mathbf{K}(\rho, \rho_i)[\mathbb{K} + \mathbb{I}\sigma_n^2]^{-1}_{ij}\mathbf{f}_j^r \\ &= \mathbf{Q}\hat{\mathbf{f}}(\rho). \end{aligned} \tag{4.3}$$

The correct behaviour of the predictive variance $\hat{\mathbf{\Sigma}}$ is also easy to check as it follows similarly from the linearity of the relevant equation (Eq. (2.12)).

To prove Property 2, first note that if the kernel function is covariant, then the transformed database $\tilde{\mathcal{D}}$ has Gram matrix $(\tilde{\mathbb{K}})_{ij} = \mathbf{K}(\mathcal{Q}_i\rho_i, \mathcal{Q}_j\rho_j) = \mathbf{Q}_i\mathbf{K}(\rho_i, \rho_j)\mathbf{Q}_j^{\mathrm{T}}$. If we define the block-diagonal matrix $\mathbb{Q}_{ij} = \delta_{ij}\mathbf{Q}_i$, this can be written in the simple block matrix form $\tilde{\mathbb{K}} = \mathbb{Q}\mathbb{K}\mathbb{Q}^{\mathrm{T}}$. Using kernel covariance again to write $\mathbf{K}(\rho, \mathcal{Q}_i\rho_i) = \mathbf{K}(\rho, \rho_i)\mathbb{Q}_{ii}^{\mathrm{T}}$ the prediction associated with the transformed database $\tilde{\mathcal{D}}$ take the form

$$\hat{\mathbf{f}}(\rho \mid \tilde{\mathcal{D}}) = \sum_{ij}^{N} \mathbf{K}(\rho, \rho_i)\mathbb{Q}_{ii}^{\mathrm{T}}[\mathbb{Q}\mathbb{K}\mathbb{Q}^{\mathrm{T}} + \mathbb{I}\sigma_n^2]^{-1}_{ij}\mathbb{Q}_{jj}\mathbf{f}_j^r. \tag{4.4}$$

By simple matrix manipulations it is now possible to show that in the above expression the symmetry transformations cancel out; indeed

$$\begin{aligned} \mathbb{Q}^{\mathrm{T}}[\mathbb{Q}\mathbb{K}\mathbb{Q}^{\mathrm{T}} + \mathbb{I}\sigma_n^2]^{-1}\mathbb{Q} &= \mathbb{Q}^{\mathrm{T}}[\mathbb{Q}(\mathbb{K} + \mathbb{I}\sigma_n^2)\mathbb{Q}^{\mathrm{T}}]^{-1}\mathbb{Q} \\ &= \mathbb{Q}^{\mathrm{T}}(\mathbb{Q}^{\mathrm{T}})^{-1}[\mathbb{K} + \mathbb{I}\sigma_n^2]^{-1}\mathbb{Q}^{-1}\mathbb{Q} \\ &= [\mathbb{K} + \mathbb{I}\sigma_n^2]^{-1}. \end{aligned} \tag{4.5}$$

The above equation, along with Eq. (4.4), implies that $\hat{\mathbf{f}}(\rho \mid \tilde{\mathcal{D}}) = \hat{\mathbf{f}}(\rho)$ as required. Moreover, the cancellation happening in Eq. (4.5) can also be used

to show that $\hat{\mathbf{\Sigma}}(\rho \mid \tilde{\mathcal{D}}) = \hat{\mathbf{\Sigma}}(\rho)$ to complete the proof of Property 2.

It is easy to check that the kernels proposed in Chapter 3 (e.g., the many-body squared exponential of Eq. (3.4) or the 2- and n-body in Eqs. (3.2) and (3.3)) do not possess the covariance property (4.2). Designing, entirely by feature engineering, a covariant kernel is in principle possible but can require complex tuning and is likely to be highly system dependent (see e.g., Ref. [30]). Note that non covariant kernels can be used and these difficulties be avoided, some having been successfully implemented [31, 77]. This leaves space for improvement as prediction efficiency will generally be enhanced by increased exploitation of symmetry (see e.g., Figure 4.3 below for a simple test of this).

4.3 Covariant integration

This section presents a general method to transform standard matrix-valued kernels into covariant ones, followed by numerical tests suggesting that the resulting kernel improves very significantly on the force-learning properties of the initial one, its error converging with just a fraction of the training data. This proceeds along the lines of previous techniques utilised in the last chapter to build symmetric scalar n-body kernels, namely the transformation integration procedure developed in Ref. [65] and used within the SOAP representation for learning potential energy surfaces of atomic systems [32, 53].

Given a group Q with elements $\mathcal{Q}$ represented by a matrix $\mathbf{Q}$ and a *base kernel* $\mathbf{K}^b$, a covariant kernel $\mathbf{K}^Q$ can be constructed by

$$\mathbf{K}^Q(\rho, \rho') = \int_Q d\mathcal{Q}_1 d\mathcal{Q}_2 \, \mathbf{Q}_1^{\mathrm{T}} \mathbf{K}^b(\mathcal{Q}_1\rho, \mathcal{Q}_2\rho')\mathbf{Q}_2 \tag{4.6}$$

where $d\mathcal{Q}$ is the normalised Haar measure for the symmetry group we are integrating over [81].

The covariance of $\mathbf{K}^Q$ as given by Eq. (4.6) is easily checked as

$$\begin{aligned} \mathbf{K}^Q(\mathcal{Q}\rho, \mathcal{Q}'\rho') &= \int d\mathcal{Q}_1 d\mathcal{Q}_2 \, \mathbf{Q}_1^{\mathrm{T}} \mathbf{K}^b(\mathcal{Q}_1\mathcal{Q}\rho, \mathcal{Q}_2\mathcal{Q}'\rho')\mathbf{Q}_2 \\ &= \int d\tilde{\mathcal{Q}}_1 d\tilde{\mathcal{Q}}_2 \, \mathbf{Q}\tilde{\mathbf{Q}}_1^{\mathrm{T}} \mathbf{K}^b(\tilde{\mathcal{Q}}_1\rho, \tilde{\mathcal{Q}}_2\rho')\tilde{\mathbf{Q}}_2\mathbf{Q}'^{\mathrm{T}} \\ &= \mathbf{Q}\mathbf{K}^Q(\rho, \rho')\mathbf{Q}'^{\mathrm{T}}, \end{aligned} \tag{4.7}$$

where the second line follows from the substitutions $\tilde{\mathcal{Q}}_1 = \mathcal{Q}_1\mathcal{Q}$ and $\tilde{\mathcal{Q}}_2 = \mathcal{Q}_2\mathcal{Q}'$. Note that these transformations have unit Jacobian because of the translational invariance (within the group) of any Haar measure [81, 82].

It can be shown that the positive semi-definiteness (Eq. (2.11)) of the base kernel is preserved under the operation (4.6) of covariant integration. In particular, a kernel is positive semi-definite if and only if it is a scalar product in some (possibly infinite dimensional) vector space [47, 83]. Hence the base kernel can be written as $\mathbf{K}^b(\rho, \rho') = \int d\alpha\, \boldsymbol{\phi}_\alpha(\rho)\boldsymbol{\phi}_\alpha^{\mathrm{T}}(\rho')$. It is then possible to show that its covariant counterpart $\mathbf{K}^Q$ (Eq. (4.6)) will also be a scalar product in a new function space. Indeed we have

$$\begin{aligned}\mathbf{K}^Q(\rho, \rho') &= \int d\mathcal{Q}_1 d\mathcal{Q}_2\, \mathbf{Q}_1^{\mathrm{T}}\mathbf{K}^b(\mathcal{Q}_1\rho, \mathcal{Q}_2\rho')\mathbf{Q}_2 \\ &= \int d\alpha\, d\mathcal{Q}_1 d\mathcal{Q}_2\, \mathbf{Q}_1^{\mathrm{T}}\boldsymbol{\phi}_\alpha(\mathcal{Q}_1\rho)\boldsymbol{\phi}_\alpha^{\mathrm{T}}(\mathcal{Q}_2\rho')\mathbf{Q}_2 \\ &= \int d\alpha\, \boldsymbol{\psi}_\alpha(\rho)\boldsymbol{\psi}_\alpha^{\mathrm{T}}(\rho'),\end{aligned} \tag{4.8}$$

where the new basis vectors were defined as $\boldsymbol{\psi}_\alpha(\rho) = \int d\mathcal{Q}\, \mathbf{Q}^{\mathrm{T}}\boldsymbol{\phi}_\alpha(\mathcal{Q}\rho)$. Hence, $\mathbf{K}^Q$ will also be positive definite.

The completely general procedure above can be cumbersome to apply in practice, because of the double integration over group elements in (4.6) and the dependence on the design of the base kernel matrix $\mathbf{K}^b$. As a simplification, we assume the base kernel to be of diagonal form; assuming equivalence of all space directions, we can then write

$$\mathbf{K}^b(\rho, \rho') = \mathbf{I}k^b(\rho, \rho'), \tag{4.9}$$

where the scalar base kernel k^b is independent on the reference frame in which the configurations are expressed. Further requiring that

$$k^b(\mathcal{Q}\rho, \mathcal{Q}\rho') = k^b(\rho, \rho'), \tag{4.10}$$

that is, scalar invariance of the base kernel upon the action of a single transformation $\mathcal{Q}$ (a property very commonly found in standard kernels), the double

integration in (4.6) reduces to a single one

$$\begin{aligned}\mathbf{K}^{Q}(\rho,\rho') &= \int d\mathcal{Q}_1 d\mathcal{Q}_2\, \mathbf{Q}_1^{\mathrm{T}}\mathbf{Q}_2 k^b(\mathcal{Q}_1\rho, \mathcal{Q}_2\rho') \\ &= \int d\mathcal{Q}_1 d\mathcal{Q}_2\, \mathbf{Q}_1^{\mathrm{T}}\mathbf{Q}_2 k^b(\rho, \mathcal{Q}_1^{-1}\mathcal{Q}_2\rho') \\ &= \int d\mathcal{Q}\, \mathbf{Q}\, k^b(\rho, \mathcal{Q}\rho'),\end{aligned} \tag{4.11}$$

where the second line follows from property (4.10) and the third line is obtained by the substitution $\mathcal{Q} = \mathcal{Q}_1^{-1}\mathcal{Q}_2$. In the next section, the integration in Eq. (4.11) is performed analytically for different base kernels. Note that incorporating prior knowledge on the correct behaviour of forces in the kernel enables learning and predicting forces associated with any configuration, regardless of its orientation. However, being able to do this for completely generic orientations is not always necessary. In many systems (e.g. crystalline solids where the orientation is known) all relevant configurations cluster around particular discrete symmetries. For these systems the relevant physics can be captured by restricting Eq. (4.11) to a discrete sum over the relevant group elements:

$$\mathbf{K}^{G}(\rho,\rho') = \frac{1}{|G|}\sum_{\mathcal{G}\in G} \mathbf{G} k^b(\rho, \mathcal{G}\rho'). \tag{4.12}$$

Since there are at most 48 distinct group elements in a crystal point group (the order of the full O_{48} group), the discrete covariant summation remains computationally feasible in bulk systems. In the particular case of one-dimensional systems, where the only symmetry operation available other than the identity is the inversion, Eqs. (4.11) and (4.12) are formally equivalent.

The procedure described, summarised by the last line of Eq.(4.11) and by Eq. (4.12), can be considered the vectorial counterpart of the more standard transformation integration procedure we have used to build scalar energy kernels in the last chapter. The last chapter has also shown that such a procedure can be avoided altogether in the case of local energy kernels as symmetric kernels can be defined directly on invariant degrees of freedom. The same line of reasoning does not hold in the case of matrix-valued covariant kernel as designing suitable covariant descriptors is arguably harder than finding invariant ones. For this reason, the automatic procedure to build covariant descriptors just described can be imagined to be comparatively more useful. In fact, co-

variant kernels built as detailed above have recently been also used recently to learn other physical tensors that are also covariant upon rotations [79, 80].

4.4 Building covariant kernels

In the previous chapter, translation and permutation invariant kernels were successfully developed (see e.g., the 2-body kernel (3.2), its n-body generalisation (3.3) and the many-body kernel (3.4)). These kernels will be used here as base kernels, and the technique developed in the previous section will be used to make those kernels covariant over the orthogonal group.

Systems with dimensions $d = 1, 2, 3$ are considered in the following three subsections. The first two provide a useful conceptual playground where the features of "covariant learning" can be more easily visualised. The third one benchmarks the method in real physical systems, simulated at the DFT level of accuracy.

1D systems

The only relevant symmetry transformation in one dimension is the reflection of a configuration through its centre. The covariant symmetrisation discussed in the previous section (Eq. (4.12)) hence takes a very simple form

$$k^{D_1}(\rho, \rho') = \frac{1}{2}(k^b(\rho, \rho') - k^b(\rho, \mathcal{F}\rho')). \tag{4.13}$$

where here and in the following C_n will denote the cyclic group of order n and D_n the dihedral group (containing also reflections) of order $2n$ (C_1 hence indicating the trivial group). Note that k^{D_1} is identically zero for inversion-symmetric configurations ρ or ρ', whose associated net force must vanish.

A key feature of covariant kernels is the ability to enable learning of the entire set of configurations that are equivalent by symmetry to those actually provided in the database. For instance, the force acting on the central atom at the origin of configuration ρ can be predicted even if only configurations ρ' of different symmetry are contained in the database. In the simplest possible system, a dimer, the only symmetry transformation maps configurations where the central atom has a right neighbour (i.e. those for which the central atom

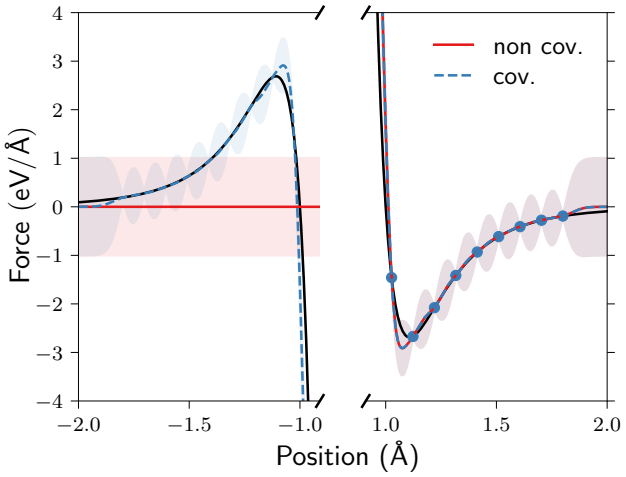


Figure 4.1: Learning the force profile of a one dimensional LJ dimer using data (blue circle) coming from one atom only. It is seen that a non covariant GP (solid red line) does not learn the symmetrically equivalent force acting on the other atom and it thus predict a zero force and maximum error. If covariance is imposed to the kernel via Eq. (4.13) (dashed blue line), then the correct equivalent (inverted) profile is recovered. Shaded regions represent the predicted standard deviation interval in the two cases.

is the left atom in the dimer) onto configurations where the central atom has a left neighbour.

The force field associated with a one dimensional Lennard Jones dimer is plotted in Figure 4.1 (black solid line) as a function of a single signed number—the 1D vector going from the central atom to its neighbour. The figure shows the predictions of an unsymmetrised 2-body base kernel using training data coming from configurations centred on the left atom only (red solid curve). This closely reproduces the true LJ forces in the region where the data are available, and predicts the pure prior mean (zero) in the symmetry related region i.e., the left half of the figure. Meanwhile, because of the covariant constraint (prior information), the GP based on the covariant kernel learns the left part of the field by just reflecting the right part appropriately.

To further check the performance improvements of the covariant symmetrisation (4.13) the above comparison is now extended to the prediction of forces associated with a one dimensional 50-atom chain system of LJ atoms in periodic boundary conditions. A database of training configurations and an independent test set of local configurations and forces were sampled from a constant temperature molecular dynamics simulation using a Langevin ther-

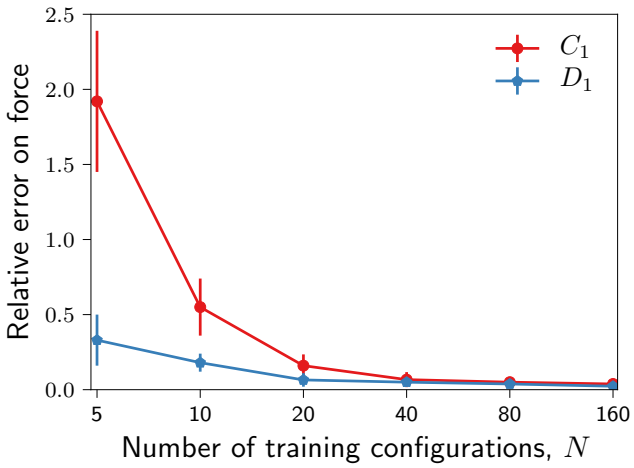


Figure 4.2: Learning curves for a one dimensional chain of LJ atoms. The covariant kernel (D_1) learns twice as fast as the base one (C_1).

mostat.

Figure 4.2 reports the average relative force error made by the GP process on the test set as a function of training set size. It is immediately apparent that the covariant kernel performance is comparable to that of the base kernel with double the amount of data points for training. We will observe the same effect also in two and three dimensions: symmetrising over a relevant finite group of order $|G|$ gives rise to an error drop approximately equivalent to a $|G|$-fold increase in the number of training points. Since the computational complexity of training GP is $\mathcal{O}(N^3)$, this can obviously lead to significant computer time savings.

2D systems

In two dimensions all rotations and reflections, as well as any combination of these, are elements of $O(2)$. This group can be represented by the following set of matrices

$$\begin{aligned} O(2) &= \{\mathbf{R}(\theta) \mid \theta \in (0, 2\pi]\} \cup \{\mathbf{R}(\theta)\mathbf{F} \mid \theta \in (0, 2\pi]\} \\ \mathbf{R}(\theta) &= \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}, \end{aligned} \tag{4.14}$$

with $\mathbf{F}$ being any 2×2 reflection matrix.

The above decomposition makes the covariant integration (4.11) over $O(2)$ trivial once the matrix elements resulting from the integration over $SO(2)$ have been calculated. We next carry out the integration for the 2-body base kernel of Eq. (3.2). This can be expressed as a sum of pair contributions, where the first atom of the pair belongs to ρ and the second to ρ':

$$\mathbf{K}_2^{SO(2)}(\rho,\rho') = \sum_{ij} \int_{SO(2)} d\mathcal{R}\, \mathbf{R}\, \mathrm{e}^{-\frac{\|\mathbf{r}_i - \mathbf{R}\mathbf{r}'_j\|^2}{2\ell^2}}. \tag{4.15}$$

Consistent with Eq. (4.11), only one atom of the pair is rotated during the integration. The pairwise integrals in (4.15) are calculated in two steps. We first define $\mathbf{R}_{ij}$ to be the rotation matrix which aligns $\mathbf{r}'_j$ onto $\mathbf{r}_i$, and then perform the change of variable $\tilde{\mathbf{R}} = \mathbf{R}\mathbf{R}_{ij}^{\mathrm{T}}$ (and equivalently, $\tilde{\mathcal{R}} = \mathcal{R}\mathcal{R}_{ij}^{-1}$) yielding

$$\mathbf{K}_2^{SO(2)}(\rho,\rho') = \sum_{ij} \left(\int_{SO(2)} d\tilde{\mathcal{R}}\, \tilde{\mathbf{R}}\, \mathrm{e}^{-\frac{\|\mathbf{r}_i - \tilde{\mathbf{R}}\mathbf{R}_{ij}\mathbf{r}'_j\|^2}{2\ell^2}} \right) \mathbf{R}_{ij}. \tag{4.16}$$

Since the two vectors $\mathbf{r}_i$ and $\mathbf{R}_{ij}\mathbf{r}'_j$ are now aligned, each integral of Eq. (4.16) can only depend on the two moduli r_i and r'_j . The final result takes a very simple analytic form (cf. Appendix A.7):

$$\mathbf{K}_2^{SO(2)}(\rho,\rho') = \sum_{ij} \mathrm{e}^{-\frac{(r_i^2 + r_j'^2)}{2\ell^2}} I_1\left(\frac{r_i r'_j}{\ell^2}\right) \mathbf{R}_{ij}, \tag{4.17}$$

where I_1 is a modified Bessel function of the first kind. The kernel in Eq. (4.17) is rotation-covariant by construction as can be seen immediately by comparison with Eq. (4.2).

By exploiting the mentioned internal structure of the orthogonal group (Eq. (4.14)), it is straightforward to show that the roto-reflection covariant kernel is given by

$$\mathbf{K}_2^{O(2)}(\rho,\rho') = \frac{1}{2}\left(\mathbf{K}^{SO(2)}(\rho,\rho') + \mathbf{K}^{SO(2)}(\rho,\mathcal{F}\rho')\mathbf{F}\right), \tag{4.18}$$

which is the two-dimensional analogue of Eq. (4.13). Interestingly, the resulting kernel can also be cast in the more intuitive form

$$\mathbf{K}_2^{O(2)}(\rho,\rho') = \sum_{ij} \mathrm{e}^{-\frac{(r_i^2 + r_j'^2)}{2\ell^2}} I_1\left(\frac{r_i r'_j}{\ell^2}\right) \hat{\mathbf{r}}_i \hat{\mathbf{r}}'^{\mathrm{T}}_j, \tag{4.19}$$

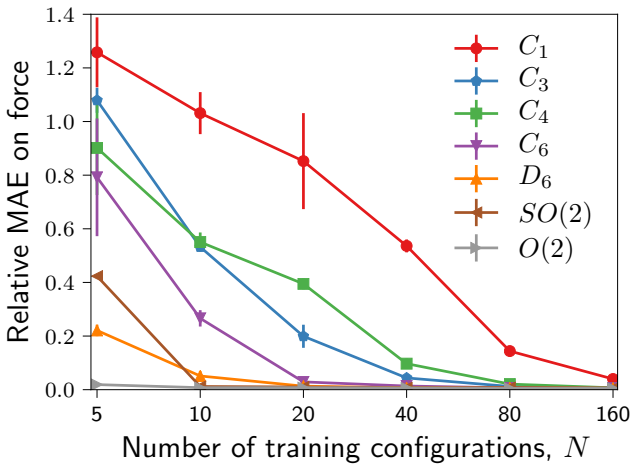


Figure 4.3: Learning curves for a 2D triangular lattice of LJ atoms. The larger the symmetry group used to construct the kernel, the faster the learning, provided that the lattice symmetry is captured.

where the hat denotes a normalised vector (cf. Appendix A.8). The above equation implies that the predicted force on an atom at the centre of a configuration ρ will be a sum of pairwise forces oriented along the directions $\hat{\mathbf{r}}_i$ connecting the central atom with all its neighbours (while each neighbour will experience a corresponding reaction force). The modulus of these forces will be a function of the interatomic distance completely determined by the training database, whose integral can be thought of as a pairwise energy potential. Clearly then, the resulting force field will be *conservative*: for any fixed database, the forces predicted by GP inference using this kernel will do zero work if integrated along any closed trajectory loop in configuration space.

Covariant integration of 3-body kernels can also be performed analytically, using a change of variable similar to that of Eq. (4.16) (cf. Appendix A.9). In this case, the resulting covariant 3-body kernels do not give rise to conservative force fields but they can give rise to much more accurate forces, which will be approximately conservative as this property is inherited by the reference quantum model.

The relative performance of the 2-body covariant kernels is tested against training and test databases sampled from a two-dimensional, 48-particle triangular lattice in periodic boundary conditions, with Lennard-Jones interactions, and kept at constant temperature via a Langevin thermostat. As the chosen

lattice has three-fold and six-fold symmetry, we can also examine the performance of covariant kernels that obey the two properties described above restricted to appropriate finite groups; these kernels are constructed as in Eq. (4.12). This allows us to monitor how imposing a progressively higher degree of symmetry on the kernel changes the rate at which forces in this system can be learned.

These results are reported in Figure 4.3. As anticipated, the discrete covariant summation over the elements of a group G is observed to be approximately equivalent to a $|G|$-fold increase of the number of data points. This can be seen e.g., from the results for the C_3 kernel (3-fold rotations) and the C_6 kernel (6-fold rotations), by comparing the error incurred in the two cases using 20 and 10 data points, respectively. More generally, we observe that the larger the group, the faster the learning. Note, however, that for the covariant summation (4.12) to extract content from the database that is actually useful for predicting forces in the test configurations at hand, the group used must describe a true underlying point symmetry of the system. Hence, for instance, the C_4 kernel gives rise to a much slower learning rate than the C_3 kernel for the 2D triangular lattice examined. Consistently, for this lattice the full point group D_6 performs almost as well as the continuous symmetry kernels, suggesting that not much more is to be gained once the full main (finite-group) symmetry of a system has been captured. This finding enables accurate force prediction in crystalline system when base kernels are used for which the covariant integration cannot be performed analytically, because the summation over a discrete symmetry group is available as a viable alternative.

3D systems

As in the two dimensional case, the covariant integration of the 2-body base kernel is here performed analytically. After expressing the integration as a sum of pairwise integrals, the position vectors $\mathbf{r}_i$ and $\mathbf{r}'_j$ of two atoms in each pair are aligned onto each other. A convenient way to achieve this is by making both vectors parallel to the z-axis with appropriate rotations $\mathbf{R}^z_i$ and $\mathbf{R}^z_j$. As before, the covariant integration will yield a matrix whose elements are scalar functions of the radii r_i and r'_j only. The integration over the rotation group ($SO(3)$ in three dimensions) can then be carried out analytically over the standard three Euler angle variables (cf. Appendix A.7 for further details).

Due to the z-axis orientation, the elements turn out to be all null except for the zz one. The result reads

$$\begin{aligned}
\mathbf{K}_2^{SO(3)}(\rho,\rho') &= \sum_{ij} \mathbf{R}_i^{z\mathrm{T}} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \phi(r_i, r'_j) \end{pmatrix} \mathbf{R}_j^z \\
\phi(r_i, r_j) &= \frac{\mathrm{e}^{-\beta_{ij}}}{\gamma_{ij}^2}(\gamma_{ij}\cosh\gamma_{ij} - \sinh\gamma_{ij}) \\
\beta_{ij} &= \frac{r_i^2 + r_j'^2}{2\ell^2} \\
\gamma_{ij} &= \frac{r_i r'_j}{\ell^2}.
\end{aligned} \tag{4.20}$$

As in the two dimensional case, the covariant kernel matrix can be rewritten only in terms of the unit vectors $\hat{\mathbf{r}}_i$ and $\hat{\mathbf{r}}'_j$ associated with the atoms of the configurations ρ, ρ' as

$$\mathbf{K}_2^{SO(3)}(\rho,\rho') = \sum_{ij} \phi(r_i, r'_j)\hat{\mathbf{r}}_i\hat{\mathbf{r}}_j'^{\mathrm{T}}, \tag{4.21}$$

making it apparent that the kernel models a pairwise conservative force field (the steps needed to show the equivalence of Eq. (4.20) and Eq. (4.21) are provided in Appendix A.8). However, while in two dimensions we needed to impose the full roto-reflection symmetry in order to obtain Eq. (4.19), rotations alone are sufficient to arrive at the fully covariant kernel in Eq. (4.21). This is a consequence of the fact that, in three dimensions, the covariant integral over rotations already imposes that the predicted force any atom will exert on any other is aligned along the vector connecting the pair: by symmetry there can be no preferred direction for an orthogonal force component after integrating over all rotations around the connecting vector, so that $\mathbf{K}_2^{O(3)} = \mathbf{K}_2^{SO(3)}$. This is not the case in two dimensions where covariant integration is over rotations around the z-axis orthogonal to all connecting vectors lying in the xy plane, so that non-aligned predicted force components associated with a non-zero torque are not forbidden by symmetry in $\mathbf{K}_2^{SO(2)}$, and only the fully symmetrised kernel (4.18) will reduce to the pairwise form (4.19). More generally we may conjecture that the rotationally covariant kernel $\mathbf{K}_2^{O(d)}$ derived from a 2-body base kernel predicts pairwise central forces, and hence is conservative, in any

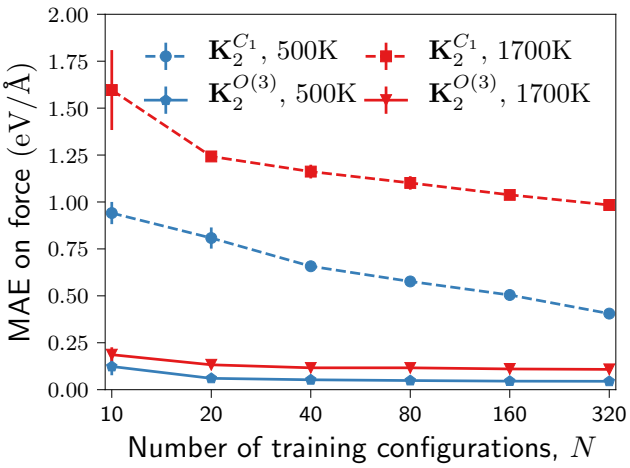


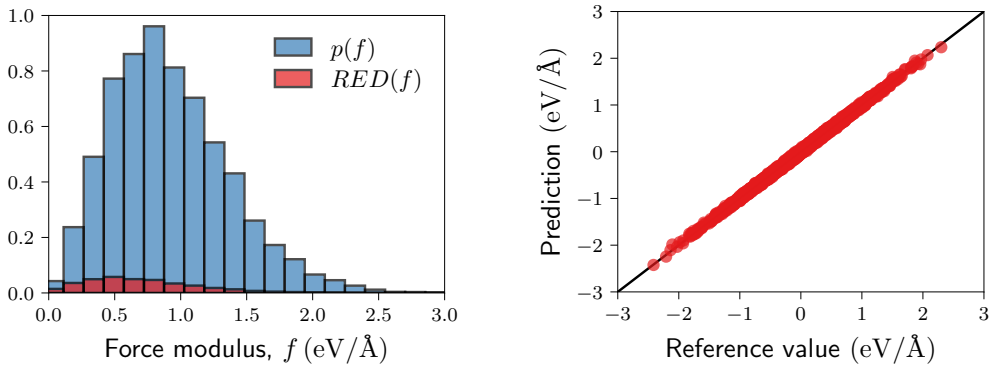
Figure 4.4: Learning Curves for crystalline nickel at two target temperatures. The $O(3)$ covariant kernel (full lines) outperforms the base one(dashed lines).

dimension d.

Note that energy conserving kernels can be obtained as double derivatives (Hessian matrices) of scalar energy kernels as described in Section 2.4. This more standard method was originally described in [54, 84] and was first used for atomistic systems in [53] to learn energies (later also used in [58] to learn forces). However, no analytic energy kernel forms exist that would yield our $O(d)$ energy conserving kernels through this route, since the double integration of the obtained kernel expressions cannot be carried out analytically.

4.5 Tests on real materials

In this section the accuracy of covariant kernels is benchmarked in predicting DFT forces in three-dimensional bulk metal systems. The test database was constructed by performing DFT-accurate dynamical simulation with exchange and correlation energy modelled via the PBE/GGA approximation [85]. The systems considered were $4 \times 4 \times 4$ supercells of fcc nickel and bcc iron in periodic boundary conditions. A weakly coupled Langevin thermostat was used to control the temperature. We first examine bulk nickel at the target temperatures of 500K and 1700K i.e., for an intermediate temperature where anharmonic behaviour is already significant, and at a temperature close to the



(a) (b)

Figure 4.5: Panel (a) shows the probability $p(f)$ of sampling a force with modulus f in a bulk of nickel atoms at $500K$ along with the relative error density $RED(f)$ made by the GP model using $N = 320$ training points. Panel (b) shows a scatter plot of reference vs. predicted force component for the same data.

melting point where the strong thermal fluctuations make the system explore a more complex target configuration space.

Figure 4.4 illustrates the performance of the kernel in Eq. (4.20) on this system. The effect of adding symmetry information on the learning curve is very significant for both temperatures. In particular, the $O(3)$ covariant kernel achieves a force error average lower than the 0.1eV/Å threshold using remarkably few training points: 10 and 80 for the lower and higher temperatures in this test, respectively. The errors of the most accurate models (achieved with a $N = 320$ database) are particularly low: $0.0435(\pm 0.0006)$eV/Å and $0.095(\pm 0.003)$eV/Å respectively. [2]

Figure 4.5 allows inspecting the accuracy of the GP predictions in a complementary way: here we plot the probability distribution of the atomic forces as a function of the force modulus (blue histogram) and the associated relative error density (grey histogram). The latter is here defined as $RED(f) = \frac{|\Delta \mathbf{f}|}{f} p(f)$, which is normalised to 0.055, reflecting the 5.5% average relative error incurred by force prediction. The fact that $RED(f)$ is everywhere a small fraction of

[2]Note that the error on each force component (often reported in the literature, and different from the error on the full force vector used here) can be expected be lower by roughly a factor $\sqrt{3}$. This yields errors of 0.025eV/Å and 0.052eV/Å in the two cases, the former comparing well with the 0.09eV/Å value obtained by using a state of the art EAM interatomic potential for nickel [26, 86].

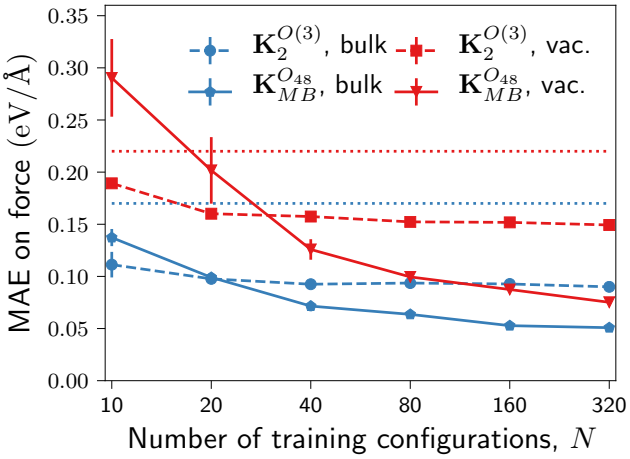


Figure 4.6: Learning curves associated with force prediction by the 2-body (dashed lines) and many-body (solid lines) covariant kernels in bulk iron systems. The dotted lines indicate the force accuracy of the EAM potential in Ref. [75] calibrated in order to match lattice constant of the reference DFT data. Blue and red colours indicate undefected systems and model systems containing a vacancy, respectively.

$p(f)$ demonstrates that a reasonable accuracy is achieved for the whole range of forces predicted.

The results presented so far indicate that fully exploiting symmetry significantly improves the accuracy of force prediction. Covariance is thus always used in the following analysis, where we compare the performance of different symmetric kernels. We start by choosing iron systems for these tests as many properties of iron-based systems remain out of modelling reach. This is mostly due to technical limitations. On the one hand full DFT calculations on large systems are too computationally expensive and even hybrid quantum-classical QM/MM simulations of iron systems are typically overwhelmingly costly, as they require large QM-zone buffered clusters to fully converge the forces [86, 87]. On the other hand, in many situations even the best available, state of the art classical force fields may not guarantee accurate force prediction, as they may incur systematic errors [86, 88], or may be hard to extend to complex chemical compositions [89], so that a technique that can indefinitely re-use all computed QM forces via GP inference and produce results that are traceably aligned with DFT-accurate forces could be very useful [30, 68].

Two bcc iron systems are here considered—both kept at constant temper-

ature of 500K with a Langevin thermostat: a 64-atom crystalline system and a 63-atom system derived from this and containing a single vacancy. In the latter, only the atoms within the first two neighbour shells of the vacancy were used to test the algorithm, to better resolve the performance of our kernels in a defective system. Figure 4.6 shows the learning curves for the two symmetrised kernels: the 2-body kernel covariant over $O(3)$ (Eq. (4.19)) and the many-body squared exponential kernel (Eq. (3.4)) made covariant over the full cubic point-group of the crystal using Eq. (4.12). The figure also reports the performance of a high-quality EAM potential [75].

The trends of this figure are very similar to those already discussed in Chaprer 3. Both kernels perform better than the EAM potentials in this test. However, the error rate of the 2-body kernel (dashed lines) levels off to some constant non-zero value that might or might not be satisfactory. The final database converged error is observed to generally depend on the system being examined and wether this will be satisfactory highly depends on the application at hand.

In bulk iron the error floor value is about 0.09eV/Å while in the vicinity of a vacancy it is considerably higher (0.15eV/Å), suggesting that in spite of its many attractive properties (e.g. fast evaluation, fast convergence, energy conservation), 2-body kernels of the form (4.21) often cannot fully capture and reproduce the reference QM physical interaction. In many situations, kernels capable of reproducing higher order interactions could be needed to reach the target accuracy. This is exemplified by the much better performance of the many-body squared exponential kernel (full lines in the figure) which yields higher accuracy, particularly for the more complex vacancy system (about 0.05eV/Å and 0.075eV/Å for atoms in the bulk and near the vacancy respectively). It is worth noting here that, in general, conserving energy exactly by construction provides no guarantee of higher force accuracy. For instance, in the case above, the squared exponential kernel delivers much more precise forces even though it conserves energy only approximately. As the approximation will in any case improve with the accuracy of the predicted forces, and with no $O(3)$-invariant energy conserving equivalent of this kernel has been proposed or appears viable, whether it is preferable to use this kernel or a less accurate but energy conserving alternative one will generally depend on both the target system and the application at hand.

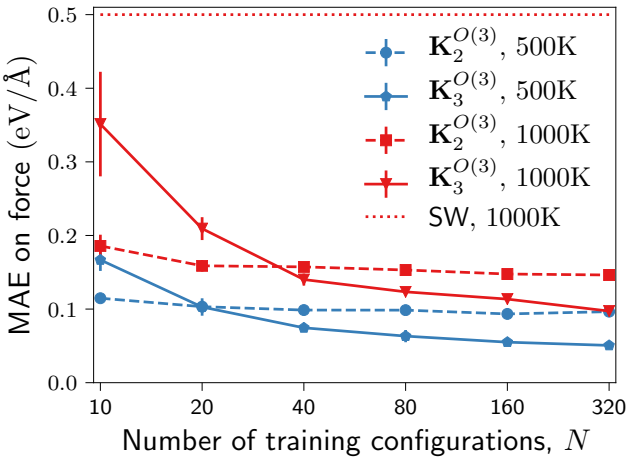


Figure 4.7: Learning curves obtained for crystalline silicon using the 2-body kernel (dashed lines) or the 3-body kernel (solid lines). Different colours indicate different temperatures.

For target systems with no clear point group symmetry, a full covariant integration would always be desirable. This cannot be carried out analytically for the squared exponential kernel, so that a discrete summation remains the only viable option for its symmetrisation.

However, interactions beyond pairwise can be still captured by e.g., the 3-body kernel. In contrast to the squared exponential kernel, this is analytically tractable and our analysis reveals that a matrix-valued 3-body kernel covariant over $O(3)$ can be derived analytically (details of the calculation can be found in Appendix A.7). The resulting model generates a roto-reflection symmetric 3-body force field that can be expected to properly describe non close-packed bonding, such as found e.g., in covalent systems.

Figure 4.7 illustrates the errors incurred by a 2- and a 3-body covariant kernel while attempting to reproduce the forces obtained during Langevin dynamics of a 64-atom crystalline silicon system, simulated using Density functional tight binding (DFTB) [90]. Both the 2- and 3-body kernels are significantly more accurate than a classical Stillinger Weber (SW) potential [23] fitted to reproduce the DFTB lattice parameter and bulk modulus [30]. As already observed in Chapter 3, due to its more restricted associated function space the 2-body kernel is the one that learns faster, and would be the more accurate if only very restricted databases had to be used. However, in this covalent system

the 3-body kernel eventually performs much better than the 2-body one for both investigated temperatures, 500K and 1000K, with errors of respectively 0.05eV/Å and 0.1eV/Å or approximately 4% and 6% of the mean force. These are very close to the minimum baseline locality error of this system, associated with the use of a finite cutoff radius r_c.

The above analysis shows that, in general, kernels of order $n > 2$ may be needed for accurate force predictions in the presence of complicated interactions, e.g. in the study of plasticity or embrittlement/fracture behaviour of covalent or metallic systems. In particular, our tests suggest that a fully $O(3)$ covariant 3-body kernel can be used successfully to improve the accuracy of force prediction in covalent materials. The results presented reveal that force covariance is achievable without imposing energy conservation to the kernel form. While both are desirable properties, lifting the exact energy conservation constraint can sometimes yield higher force accuracy. For instance, no invariant local energy based kernel has been proposed for the squared exponential ("universal approximator") kernel, since the analytic integration over $O(3)$ is not viable. However, we find that covariance limited to the O_{48} point group is very effective for force predictions in crystalline Fe systems using this kernel (see Fig. 4.6). In general, while predicting forces with high accuracy is the main motivation for machine learning-based work in this field, the best compromise between accuracy, energy conservation and covariance will depend on the specific target application. For instance, kernels built from a covariant integration (or summation) that do not conserve energy exactly should not be used as substitutes for conventional interatomic potentials to perform long constant energy simulations, since they might in principle lead to non-negligible spurious energy drifts. This is not a problem in constant temperature simulations, where a thermostat exchanging energy with the system will be able to compensate for any such drift if appropriately chosen [91]. Furthermore, the same kernels will be particularly suited for schemes that are in all cases incompatible with strict energy conservation. These include the LOTF approach and any online learning scheme similarly involving a dynamically updated force model. They also include any scheme based on a fixed but very large database where, to maximise efficiency, each force prediction only uses its corresponding most relevant database subset. On the other hand, any usage style is possible for covariant kernels conserving energy exactly, such as

the covariant 2-body kernels of Eqs. (4.13), (4.19), and (4.21). In fact, the conservative pairwise interaction forces generated by these covariant 2-body kernels can be easily integrated to provide effective standard pairwise potentials for any application needing a total energy expression. We also note that while the pair interaction form would still ensure very fast evaluation of the predicted forces, its accuracy for complex systems could be improved by dropping the transferability requirement of a single pairwise function. In such a scheme, different system regions could conceivably be modelled by locally optimised forces/potentials, where the local tuning could be simply achieved by restricting the inference process to subsets of the database pertinent to each target region. This approach is discussed in more details in Section 5.4 of the next chapter.

4.6 Summary

This chapter described a novel method to learn quantum forces on local configurations. This method is based on a vectorial GP and on the inclusion of prior knowledge in a matrix-valued kernel function. Section 4.2 showed how to include rotation and reflection symmetry of the force in the GP model via the notion of covariant kernels. Section 4.3 provided a general recipe to impose this property on otherwise non-symmetric kernels. The essence of this recipe lies in a special integration step, which was named covariant integration, over the full roto-reflection group associated with the relevant number of system dimensions. In Section 4.4, this calculation was performed analytically starting from a 2-body or a 3-body base kernel. When 2-body kernels are made covariant, the resulting $O(d)$ covariant kernels can be shown to generate conservative force fields.

We furthermore tested covariant kernels on standard physical systems in one, two and three dimensions. The one and two dimensional scenarios served as playgrounds to better understand and illustrate the essential features of such learning. The three dimensional systems (discussed in Section 4.5) allowed some practical benchmarking of the methodology in real systems. In agreement with what physical intuition would suggest, incorporating symmetry consistently gave rise to more efficient learning. Moreover, it was found that if both database and target configurations belong to a system with a def-

inite underlying symmetry, restricting kernel covariance to the corresponding finite symmetry group delivers the full speed-up of error convergence with respect to database size. At the same time this approach lifts the requirement of analytical integrability over the full $O(d)$ manifold, as the restricted integration becomes a simple discrete sum over the relevant finite set of group elements. Testing on nickel, silicon and iron (the latter both pure and defective) reveals that the present recipes can improve significantly on available classical potentials.

Chapter 5

Selecting the best model

5.1 Introduction

Through the previous chapters, this thesis has introduced several GP models to infer a classical force field starting from a dataset of quantum calculations. These differed in the type of kernel function used which, to give some examples, could be one of the n-body energy kernels proposed in Chapter 3 or one of the covariant matrix-valued kernels of Chapter 4. Many more kernels for energies and forces have been proposed in the literature, some of the most notable ones being those based on the SOAP [32], the Coulomb matrix [55] or the bag of bonds representation [92]. Moving out of the realm of GP regression, other fitting algorithms have been proposed based on neural networks [15], generalised linear models [93], not to mention the many available classical parametrised models. With the above list of methods being surely incomplete, the problem of selecting a single model is both interesting and unavoidable.

The No free lunch theorems proven by D. H. Wolpert in 1996 state that no learning algorithm can be considered better than any other (and than random guessing) in a general sense [94]. This remarkable result seems to suggest that the best among competing models has to be chosen in relation to the particular system studied and the dataset available. An attractive approach lies in the long-standing Occam's razor principle i.e., select the simplest model that is still able to provide a satisfactory explanation [95–97]. Hence, in the context of force field learning, one should incorporate as much prior knowledge as is available on the function to be learned and the particular system at hand. When prior knowledge is not enough to decide among competing models, these

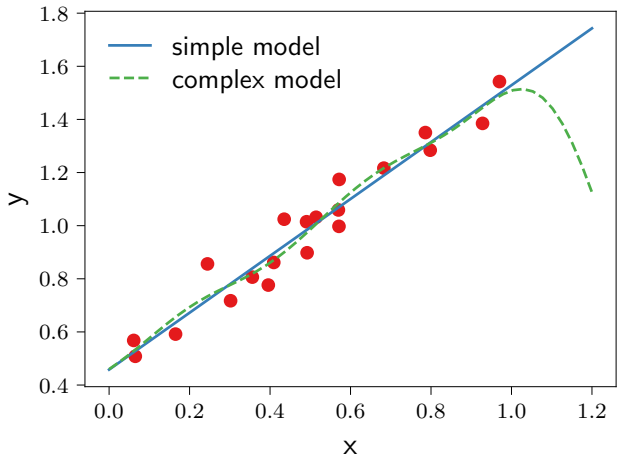


Figure 5.1: A simple linear model (blue solid line) and a complex GP model (green dashed line) are fitted to some data points. In this situation, if we have prior knowledge that a linear trend underpins the data, we should enforce the blue model *a priori*; otherwise we should select the blue model by Occam's razor, since it is the simplest one. The advantages of this choice lie in the greater interpretability and extrapolation power of the simpler model.

should all be trained and tested, after which the simplest one that is still compatible with the desired target accuracy should be selected. This approach is illustrated in Figure 5.1, where two competing models are considered for a one dimensional data set.

Such a principle has driven the heuristic based method of model selection suggested by the analysis of Chapter 3 (Section 3.4), where the best model could be considered to be the simplest kernel compatible with a given target accuracy. However, the mentioned method is not theoretically sound and fails when the set of kernels (or more generally the set of models) considered cannot be clearly ranked in terms of their complexity.

This chapter details how the theory of Bayesian model selection embodies the Occam's razor principle and how it can provide a rigorous approach to the selection of force fields models, specifically GP-based ones. Section 5.2 deals with the general theory of Bayesian model selection and its relation to the Occam's razor principle. Section 5.3 details the specific way in which the general theory can be applied in practice for selecting the order n of an n-body kernel. The analysis, carried out in either one or three dimensional systems, reveals

that low order kernels are advantageous to use, not only for physical systems of low complexity but for any system when the available database is too small to resolve the complexity of the underlying interaction. The advantages of choosing a low order model also include more physically driven reasons, these are illustrated in Section 5.4, where some ideas for circumventing the representation power limitations of a low order force field are also presented.

5.2 Theory of Bayesian model selection

Let us assume that we want to select a single model out of the set $\{\mathcal{M}_n^{\boldsymbol{\theta}}\}$ (each e.g., defined by a kernel function of given order n). Each model is equipped with a vector of *hyperparameters* $\boldsymbol{\theta}$, (in our context this will be associated with the covariance lengthscale ℓ, the data noise level σ_n, and similar). The model one should select in a Bayesian framework is that with the largest posterior probability, conditioned on a given set of reference calculations $\mathcal{D} = \{(\varepsilon_i^r, \rho_i)\}_{i=1}^N$. This probability can be formally written down using Bayes' theorem as

$$p(\mathcal{M}_n^{\boldsymbol{\theta}} \mid \boldsymbol{\rho}, \boldsymbol{\varepsilon}^r) = \frac{p(\boldsymbol{\varepsilon}^r \mid \boldsymbol{\rho}, \mathcal{M}_n^{\boldsymbol{\theta}}) p(\mathcal{M}_n^{\boldsymbol{\theta}})}{p(\boldsymbol{\varepsilon}^r \mid \boldsymbol{\rho})}. \tag{5.1}$$

However, often little *a priori* information is available on the candidate models and their hyperparameters (or it is simply interesting to operate a selection unbiased by priors, and "let the data speak"). In such a case, the prior $p(\mathcal{M}_n^{\boldsymbol{\theta}})$ can be ignored as being flat and uninformative, and maximising the posterior becomes equivalent to maximising the *marginal likelihood* $p(\boldsymbol{\varepsilon}^r \mid \boldsymbol{\rho}, \mathcal{M}_n^{\boldsymbol{\theta}})$ (here equivalent to the *model evidence* [1]), and the optimal selection tuple $(n, \boldsymbol{\theta})$ can be hence chosen as

$$(\hat{n}, \hat{\boldsymbol{\theta}}) = \underset{(n,\boldsymbol{\theta})}{\operatorname{argmax}}\ p(\boldsymbol{\varepsilon}^r \mid \boldsymbol{\rho}, \mathcal{M}_n^{\boldsymbol{\theta}}). \tag{5.2}$$

The marginal likelihood can be computed analytically in the case of GP models, being the normalised multivariate distribution given in Eq. (2.6).

[1] The model evidence is conventionally defined as the integral over the hyperparameter space of the marginal likelihood times the hyperprior (cf. [47]). We here simplify the analysis by jointly considering the model and its hyperparameters.

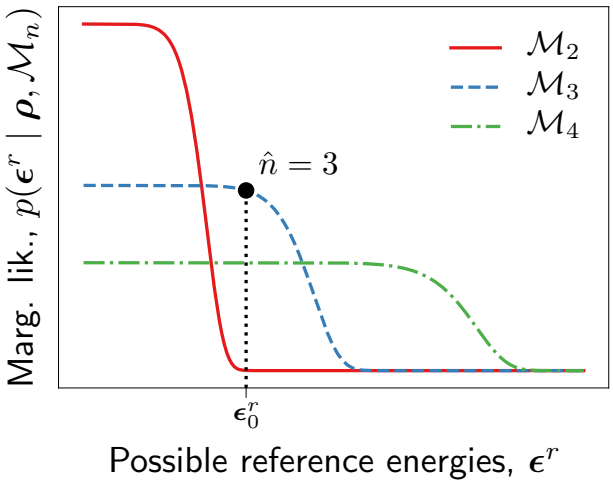
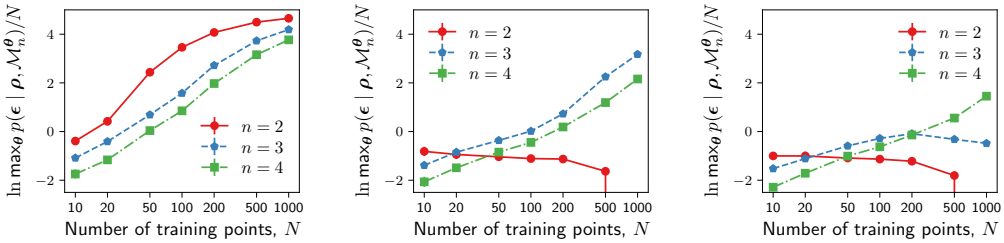


Figure 5.2: Cartoon of the marginal likelihood profile of three models of increasing complexity. More complex models can fit very different datasets $\boldsymbol{\varepsilon}^r$, this is illustrated by the fact that their marginal likelihood is non-zero for a broader region of the dataset space (here pictorially one dimensional).

The maximisation in Eq. (5.2) can be thought of as a Bayesian formalisation of the Occam's razor principle. This is illustrated in Figure 5.2, which contains a cartoon of the marginal likelihood of three models of increasing complexity/flexibility (a useful analogy is to think of polynomials $P_n(x)$ of increasing order n, the likelihood representing how well these would fit a set of measurements $\boldsymbol{\varepsilon}^r$ of an unknown function $\varepsilon(x)$). By definition, the most complex model in the figure is the green one, as it assigns a non-zero probability to the largest domain of possible outcomes, and would thus be able to explain the widest range of datasets. Consistently, the simplest model is the red one, which is instead restricted to the smallest dataset range (in our analogy, a parabola will be able to fit well fewer data sets than a fourth order polynomial). Once a reference database $\boldsymbol{\varepsilon}_0^r$ is collected, it is immediately clear that the $\mathcal{M}_3$ model with highest likelihood $p(\boldsymbol{\varepsilon}^r \mid \boldsymbol{\rho}, \mathcal{M}_n^{\boldsymbol{\theta}})$ at $\boldsymbol{\varepsilon}^r = \boldsymbol{\varepsilon}_0^r$ is the simplest that is still able to explain it (the blue one in Figure 5.2). Indeed, the even simpler model $\mathcal{M}_2$ is not likely to explain the data, the more complex model $\mathcal{M}_4$ can explain more than is necessary for compatibility with the ε_0^r data at hand, and thus produces a lower likelihood value, due to normalisation.



(a) $n^t = 2$ (b) $n^t = 3$ (c) $n^t = 4$

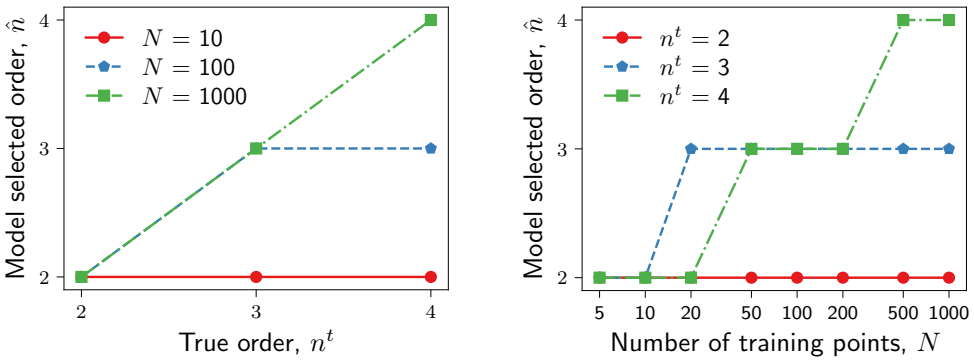
Figure 5.3: Scaled log maximal marginal likelihood as a function of the number of training points N for different kernel orders n and true interaction orders n^t.

5.3 Model selecting n-body kernels

The model selection methodology just described could be in principle used to select the best within any set of GP kernel, which might contain indistinctly scalar n-body kernels, matrix-valued covariant kernels as well as any other kernel not treated in this thesis [30–32, 55, 92]. This section however focuses on the restricted but representative class of scalar n-body kernels and tests Bayesian model selection on the problem of selecting the order n given a set of target calculations. It is instructive to first analyse a simple one dimensional system with controllable interaction order, while real three dimensional materials are analysed afterwords.

1D systems

The system considered here is a one dimensional chain of atoms interacting via an *ad hoc* potential of order n^t (t standing for "true") (see Appendix A.10 for more details on this model). For each value of n^t, a database was generated by random sampling of N configurations and associated energies and the corresponding optimal lengthscale parameter $\hat{\ell}$ and interaction order $\hat{n}$ of the n-kernel in Eq. (3.3) were found by solving the maximisation problem of Eq. (5.2). This procedure was repeated 10 times to obtain statistically significant conclusions, the results were however found to be very robust in as much as they they were found not to depend significantly on the specific realisation of the training dataset.

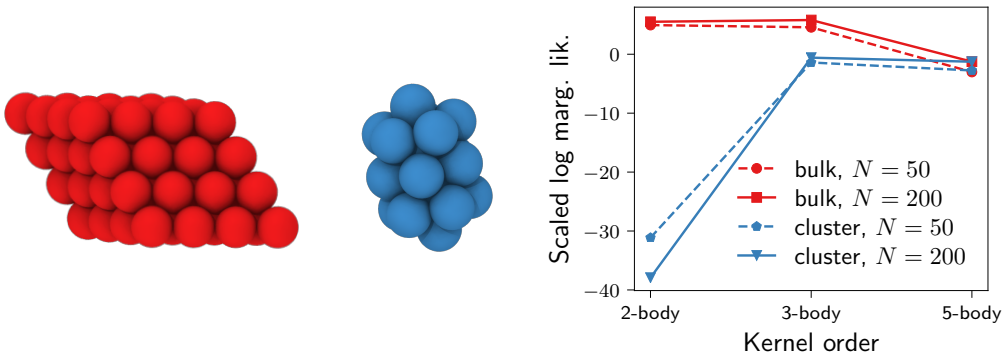


(a) (b)

Figure 5.4: Model selected order $\hat{n}$ as a function of the true order n^t (left) and as a function of the number of training data points N (right).

The results are reported in Figure 5.3, which contains the logarithm of the marginal likelihood maximised over the hyperparameter ℓ, divided by the number of training points N, as a function of N for different combinations of true orders n^t and kernel order n. The model selected in each case is the one corresponding to the line achieving the maximum value of this quantity. It is interesting to notice that, when the kernels order is lower than the true order (i.e., for $n < n^t$), the maximal marginal likelihood can be observed to *decrease* as a function of N (as e.g., the red and blue lines in Figure 5.3(c)). This makes the gap between the true model and the other models increase substantially as N becomes sufficiently large.

Figure 5.4 summarises the results of model selection. In particular, Figure 5.4(a) illustrates the model-selected order $\hat{n}$ as a function of the true order n^t, for different training set sizes N. The graph reveals that, when the dataset is large enough ($N = 1000$ in this example) maximising the marginal likelihood always yields the true interaction order (green line). On the contrary, for smaller database sizes, a lower interaction order value n is selected (blue and red lines). This is consistent with the intuitive notion that smaller databases may simply not contain enough information to justify the selection of a complex model, so that a simpler one should be chosen. More insight can be obtained



(a) (b)

Figure 5.5: Panel (a): the two nickel systems used in this section as examples, with bulk fcc nickel in periodic boundary conditions on the left (red) and a nickel nanocluster containing 19 atoms on the right (blue). Panel (b): maximum log marginal likelihood for the 2-, 3- and 5-body kernels in the bulk Ni (red) and Ni nanocluster (blue) systems, using 50 (dashed lines) and 200 (full lines) training configurations.

by observing Figure 5.4(b), reporting the model selected order as a function of the training dataset size for different true interaction orders. While the order of a simple 2-body model is always recovered (red line), to identify as optimal a higher order interaction model a minimum number of training points is needed, and this number grows with the system complexity.

3D systems

The maximal marginal likelihood principled is here used to select the optimal kernel model for two DFT nickel systems. The first is an undefected fcc crystal in periodic boundary conditions (Figure 5.5(a) left) kept at a temperature of 500K via a Langevin thermostat. The second is a nanocluster of 19 atoms (Figure 5.5(a) right) simulated at 300K. More details on the datasets used are available in Appendix A.6). Differently from the one dimensional toy model just discussed, here there is no notion of "true" interaction order as a quantum interaction is theoretically always a many-body one. However, there certainly is a notion of complexity of the physical interactions occurring, and more complex systems will require a model of higher interaction order. One

can expect that, in spite of the higher temperature fluctuations of the crystal, the greater surface effects of the nanocluster could make the latter still the most complex system and consequently the one for which a higher order is selected.

The set of models considered for this test $\{\mathcal{M}_2^{\boldsymbol{\theta}}, \mathcal{M}_3^{\boldsymbol{\theta}}, \mathcal{M}_5^{\boldsymbol{\theta}}\}$ comprises the 2-body kernel (3.15), the 3-body kernel (3.15), and the 5 body kernel (3.18). The hyperparameter vectors comprise prior noise σ_n and lenghtscale ℓ for each candidate model $\boldsymbol{\theta} = (\sigma_n, \ell)$. This list will suffice for our purposes but is by no means a comprehensive list of models and hyperparameters one can envision for the systems under consideration. For instance, one could generate more expressive models by additively mixing n-body kernels (as also done e.g., in Ref. [61]). In particular, mixing a 3-body kernel and a 2-body kernel having a larger cutoff radius would allow the latter to capture also longer range interactions—which could be modelled accurately with a simple pairwise potential. The weight in front of each kernel in the series (and potentially also the cutoff radii of the kernels) can be treated as hyperparameters and can in principle be optimised by maximising the marginal likelihood.

In the present experiment, similarly to what was done for the one dimensional system, the noise hyperarameter σ_n was kept fixed to what was *a priori* believed to be the intrinsic locality error of the forces [2], while the lengthscale parameter ℓ was instead optimised separately for each kernel and training set in order to maximise the marginal likelihood. Figure 5.5(b) shows the results obtained in the form of a graph of the maximal marginal likelihood as a function of kernel order, for the two materials and for training sets of either 50 or 200 configurations. For the crystalline nickel system (red lines) the marginal likelihood has a maximum at interaction order $n = 2$ for both $N = 50$ (dahsed line) and for $N = 200$ (solid line). This points at the optimality of a 2-body kernel for closed packed undefected nickel systems. Clearly, being able to reproduce forces in in the crystalline phase does not guarantee accuracy for the same material in other circumstances. In fact, in the case of the nickel nanocluster (blue curves) the value of the maximal marginal likelihood of a 3-body kernel is slightly higher than the 5-body one and substantially higher than the 2-body one, pointing at the inappropriateness of 2-body modelling for such

[2] Alternatively, the noise hyperparameter could be set to represent the target accuracy needed for a given application. This would make the maximum marginal likelihood principle automatically select the simplest model compatible with the chosen level of uncertainty.

a system and at the optimality of a 3-body kernel. Importantly, the 3-body model is selected for the nanocluster system even for the small training set ($N = 50$). This shows that the maximum marginal likelihood principle is able to correctly identify the minimum interaction order needed for a fundamental characterisation of a material even with very moderate training set sizes. This minimum order will generally depend on the nature of the chemical interactions involved, but for most inorganic material it can be expected to be low (either 2 or 3 as a consequence of the ionic or covalent nature of the chemical bonds involved).

The results presented so far suggest that lower order (simpler) models are often selected over more complex ones by the maximum marginal likelihood principle, especially when the size of the training database is not sufficient to fully resolve higher oder interactions. Although not immediately obvious, choosing a simpler model typically also leads to smaller prediction errors on unseen configurations, since overfitting is ultimately prevented as clear from e.g., the numerical tests shown in Chapter 3 (Figure 3.5).

5.4 The advantage of low order models

The picture emerging from the observations made so far is one in which, although the quantum interactions occurring in atomistic systems will in principle involve all atoms in the system, there is never going to be sufficient data to select/justify the use of interaction models beyond the first few terms of the many-body expansion (or any similar expansion based on prior physical knowledge). Furthermore, using low order models presents strong advantages which cannot be ignored even when very large datasets are available. Indeed, putting aside their greater interpretability (which however can be very useful for physically based validation), low order models very unlikely undergo overfitting, and they hence naturally generalise better in unexplored regions of configuration space (see e.g., the discussion in Ref. [61]).

At the same time, in many likely scenarios, a realistic target threshold for the average error on atomic forces (say of the order of 0.1eV/Å) will be met by truncating the series at a complexity order that is still computationally manageable. Hence, in practice *a small finite order model will always be optimal*.

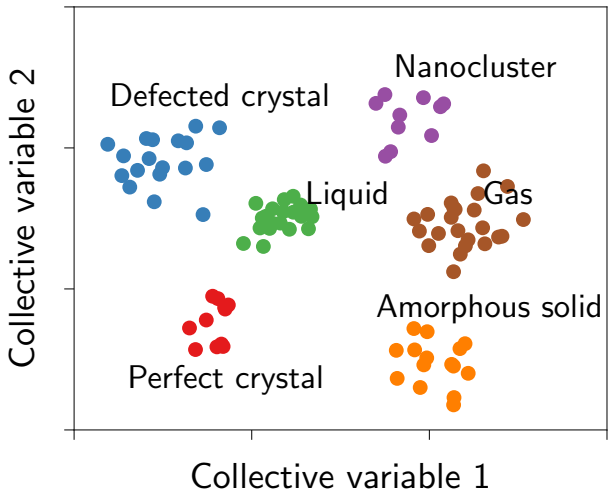


Figure 5.6: A an illustrative representation of a heterogeneous database composed of configurations which "cluster" around specific centroids in an arbitrary two dimensional space. The different clusters can be imagined to be different phases of the same material.

This is in stark contrast with the original hope of finding a single many-body "universal approximator" model to be used in every context, which has been driving a lot of interest in the early days of the ML-FF research field, producing for instance the reference methods [15, 16]. Furthermore, the observation that it may be possible to use models of finite-order complexity without ever recurring to universal approximators suggests alternative routes for increasing the accuracy of GP models without increasing the kernels' complexity. These are worth a small digression.

Imagine a situation as the one depicted in Figure 5.6, where we have an heterogeneous dataset composed of configurations that cluster into groups. This could be the case, for instance, if we imagine collecting a database which includes several relevant phases of a given material. Given the large amount of data and the complexity of the physical interactions within (and between) several phases, we can imagine the model selected when training on the full dataset to be a relatively complex one. On the other hand, each of the small datasets representative of a given phase may be well described by a model of much lower complexity. As a consequence, one could choose to train several GPs, one for each phase, as well as a *gating function* $p(c \mid \rho)$ deciding, during an MD run, which of the clusters c to call at any given time. Each GP learner

will effectively *specialise* on a particular phase of the material. This model can be considered a type of *mixture of experts* model [98, 99], and heavily relies on a viable partitioning of the configuration space into clusters that will comprise similar entries. This subdivision is far from trivially obtained in general, and in fact obtaining "atlases" for real materials or molecules similar to the one in Figure 5.6 is an active area of research [18, 20, 100, 101]. Another technique to combine multiple learners could be that of bootstrap aggregating ("Bagging") [102]. In our particular case, this could involve training multiple GPs on random subsections of the data and then averaging them to obtain a final prediction. While it should not be expected that the latter combination method will perform better than a GP trained on the full dataset, the approach can be very advantageous from a computational perspective since, similar to the mixture of experts model, it circumvents the $\mathcal{O}(N^3)$ computational bottleneck of inverting the kernel matrix in Eq. (2.7) by distributing the training data to multiple GP learners. The algorithms described above and in general ML algorithms based on the use of multiples learners belong to a broader class of *ensemble learning* algorithms [103, 104].

5.5 Summary

Motivated by the need of choosing among competing models—including e.g, those developed in Chapter 3 and 4 as well as other potential candidates from the the literature—this chapter was dedicated to the theory of Bayesian model selection and its application to choosing an optimal kernel within a set of potential candidates. After the necessary background was introduced, leading to the principle of maximum marginal likelihood within GP regression, model selection was carried out over the set of n-body kernels introduced in Chapter 3. The procedure was first exemplified in a one dimensional toy model of controllable interaction order and later applied to realistic DFT systems of nickel atoms. In both cases, it was found that low order models were selected not only when the systems' interaction was actually of low order, but also when only insufficiently large training datasets are available, not containing enough information to fully resolve more complex interactions.

Choosing a low order ("simple") model guarantees that the corresponding GP does not overfit to the data and generalises well to unexplored regions of

configuration space, not present in the training database, making the resulting force field more robust and transferable. Furthermore, simple models are more easily interpreted and validated, in as much as the trained force field can be readily visualised and examined.

In spite of the many attractive features of low order models, they suffer from obvious limitations regarding their flexibility when compared to e.g., universal approximators. To circumvent this problem, some recipes based on ensemble learning ideas where suggested.

Chapter 6

Speeding up low-n models

6.1 Introduction

Perhaps contrary to expectations, but perfectly in line with the lesson learned from parametrised force fields, this thesis has shown that models of low interaction oder are often optimal both in the Bayesian sense of having maximal evidence (Chapter 5) and in the practical sense of providing the lowest error on unseen configurations (Chapters 3, 4 and 5). The reasons for their success were identified in the greater transferability and interpretability that low order models offer when compared to higher or infinite order ones.

This chapter will further show that these models provide a very substantial advantage also in terms of evaluation time. Indeed, in spite of being orders of magnitude faster than DFT calculations, GPs remain considerably slower than standard parametrised force fields. This *does not* need to be the case when low-n kernels are used and this chapter explains how this computational gap can be bridged. In particular, a "mapping" procedure is introduced and tested, consisting in the storage and local interpolation the learned energy profile on a grid of points on the relevant degrees of freedom of the n-body interaction.

Section 6.2 illustrates the idea behind mapping taking the first non-trivial interaction order $n = 3$ as an example. In Section 6.3 the procedure is tested on iron and silicon systems, confirming that very substantial computational gains can be achieved.

6.2 Mapped force fields

It is clear that once a GP kernel is recognised as being n-body, it automatically defines an n-body force field corresponding to it, for any given choice of training set. This will be an n-body function of atomic positions, whose values *can* be computed by GP regression sums over the training set as done by standard ML-FF implementations, but *do not have to* be computed this way. In particular, the execution speed of a machine learning-derived n-body force field might be expected to depend on its order n (e.g., for $n = 3$ it will involve sums over all atomic triplets, like any 3-body parametrised force fields), but should otherwise be independent of the training set size.

It is therefore possible to construct a mapping procedure yielding a machine learning-derived, nonparametric force field (here called "MFF") that allows a very significant speedup over calculating forces by direct GP regression. For convenience, the first non-trivial interaction order $n = 3$ is here considered. It is first shown that a 3-body GP exactly corresponds to a classical 3-body MFF, and later explained how the mapping yielding the MFF can be carried out, in this case using a 3D-spline approximator.

To explicitly compute the the 3-body force field, implicitly defined by a GP with a 3-body kernel, we start by writing the GP predictive mean as

$$\hat{\varepsilon}(\rho) = \sum_{d=1}^{N} \sum_{\substack{i_1 > i_2 \in \rho \\ j_1 > j_2 \in \rho_d}} \tilde{k}_3(\mathbf{q}_{i_1,i_2}, \mathbf{q}^d_{j_1,j_2})\alpha_d, \tag{6.1}$$

where the general form of a 3-body kernel (Eq. (3.16)) was used. Then, by inverting the order of the sums over the database and atoms in the target configurations, we obtain the explicit expression for the otherwise implicit 3-body potential:

$$\begin{aligned} \hat{\varepsilon}(\rho) &= \sum_{i_1 > i_2 \in \rho} \tilde{\varepsilon}(\mathbf{q}_{i_1,i_2}) \\ \tilde{\varepsilon}(\mathbf{q}_{i_1,i_2}) &= \sum_{d=1}^{N} \sum_{j_1 > j_2 \in \rho_d} \tilde{k}_3(\mathbf{q}_{i_1,i_2}, \mathbf{q}^d_{j_1,j_2})\alpha_d. \end{aligned} \tag{6.2}$$

The above equation reveals that the GP defines the local energy of a configuration as a sum over all triplets containing the central atom, where the

function $\tilde{\varepsilon}(\mathbf{q}_{i_1 i_2})$ represents the energy associated with each triplet $\mathbf{q}_{i_1 i_2} = (r_{i_1}, r_{i_2}, r_{i_1 i_2})^{\mathrm{T}}$. The triplet energy is calculated by three nested sums, one over the N database entries and two running over the M atoms of each database configuration (M may slightly vary over configurations, but can be assumed to be constant for the present purpose). The computational cost of a single evaluation of the triplet energy $\tilde{\varepsilon}$ in Eq. (6.2) scales consequently as $\mathcal{O}(NM^2)$. Clearly, improving the GP prediction accuracy by increasing N and M will make the prediction slower.

However, such a computational burden can be avoided, bringing the complexity of calculating the triplet energy $\tilde{\varepsilon}$ in Eq. (6.2) to $\mathcal{O}(1)$. Since the triplet energy $\tilde{\varepsilon}$ is a function of just three variables (the effective symmetry-invariant degrees of freedom associated with three particles in three dimensions), we can calculate and store its values on an appropriately distributed grid of points within its domain.

This procedure effectively maps the GP predictions on the relevant 3-body feature space: once completed, the value of the triplet energy at any new target point can be calculated via a local interpolation, using just a subset of nearest tabulated grid points. If the number of grid points N_g is made sufficiently high, the mapped function will be essentially identical to the original one but, by virtue of the locality of the interpolation, the cost of evaluating it will not depend on N_g.

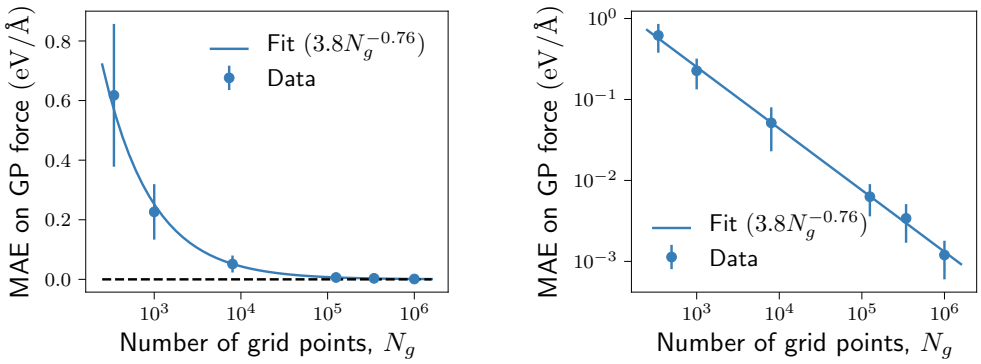
In practice, a spline interpolation of the so-tabulated potential can be very easily used to predict any $\hat{\varepsilon}$ or its negative gradient $\hat{\mathbf{f}}$ (analytically computed, to allow for a constant of motion in MD runs). The interpolation approximates the GP predictions with arbitrary accuracy, which increases with the number of points N_g in the grid of tabulated values, as illustrated in next section.

The formal generalisation of the developed technique to any finite order n is straightforward provided that a good interpolator can be identified and implemented. The computational speed of the resulting MFF can similarly be expected to be independent of the number of training points N and to depend linearly on the number of distinct atomic n-plets present in a typical atomic environment ρ including M atoms plus the central one (this is the number of combinations $\binom{M}{n-1} = M!/(n-1)!(M-n+1)!$, yielding e.g., M pairs and $M(M-1)/2$ triplets). This would result in an overall $N\binom{M}{n-1}$ speedup factor against a GP using an n-body kernel given by Eq. (3.16), as this would instead

scale linearly with N and quadratically with the number of n-plets present in an environment.

In practice however, there are two major limitations to using this approach for $n > 3$. Firstly, while mapping 2- or 3-body predictions on a one or three dimensional spline is straightforward, the number of values to store using a regular grid of points grows exponentially with n, consistent with the rapidly growing dimensionality associated with atomic n-plets. This makes the procedure very challenging for higher n values which would require ($3n$-6)-dimensional mapping grids and interpolation splines. Secondly, one should note that the evaluation time of non-unique n-body kernels obtained as powers of an n'-body input kernel do not scale as the number of n-plets but only with the number of n'-plets, independently on n (as described in Section 3.3). This means that in practice, high order GP kernels can be built as powers of a 3-body kernel as done in Section 3.3 and the corresponding high order MFF would quickly become slower to evaluate than the original GP.

On a brighter note, flexible 3-body force fields were shown to capture most of the features for a variety of materials [43, 60, 61, 105]. Increasing the order of the kernel function beyond three might be unnecessary for many systems (and it could actually be advantageous to use a low-n model as discussed in Chapters 3,4 and 5). Hence, building extremely fast yet flexible and accurate 3-body force fields could represent a "sweet spot" for many practical applications. Moreover, like their GP counterparts, and unlike parametric force fields, MFFs potentially offer a natural measure of uncertainty that could be used to monitor whether any extrapolation is taking place that might involve large prediction errors: the GP predicted variance $\hat{\sigma}^2$. This can in principle also be mapped. However, since the predictive variance depends on *couples of n-plets* its exact mapping is rather cumbersome already for $n > 2$. For instance, for $n = 3$, mapping the predictive variance exactly would mean storing a function of $3 + 3 = 6$ variables providing the covariance of each triplet with any other one in the target configurations. A very simplifying assumption would involve treating all the triplets as independent (zero cross-covariance) and only store the three dimensional function providing the variance of each triplet singularly (the interested reader is referred to Appendix A.11 for details on the whole procedure). While this alternative measure would probably not correlate well with the original GP variance (in as much as the cross-covariance terms will



(a) (b)

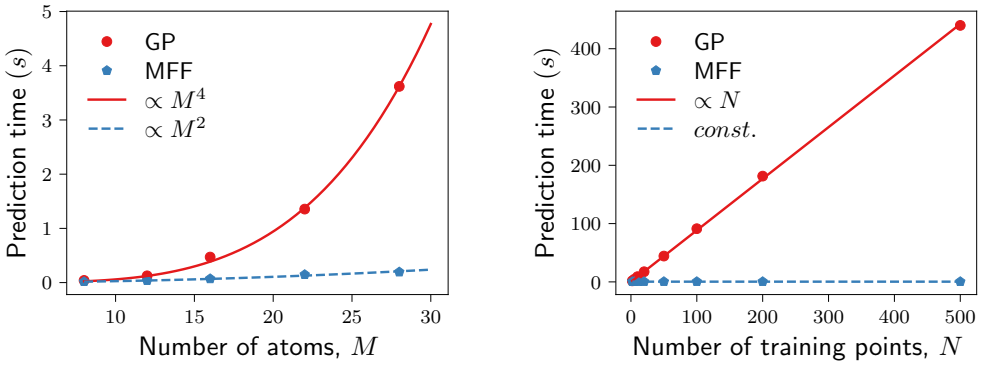
Figure 6.1: Error made by the MFF on GP forces predicted using the 3-body GP kernel in Eq. (3.15) as a function of the number of grid points N_g used for the spline interpolation. The MFF was constructed on distances between 1.5Å and 4.5Å. Panels (a) and (b) show the same data on a linear-log or log-log scale respectively.

generally *not* be zero), it still represents a valid and well grounded measure of uncertainty, and it remains to be investigated how well it can predict extrapolative scenarios.

6.3 Tests on real materials

The production and use of MFFs is here tested for two materials: Crystalline iron in the presence of a vacancy and amorphous silicon, both in periodic boundary conditions (cf. Appendix A.6 for details on the datasets).

Figure 6.1 shows the convergence the mapped forces derived from the 3-body kernel in Eq. (3.15), for the iron database. The interpolation is carried out using a three dimensional cubic spline for different mesh sizes. Comparison with the reference forces produced by the GP allows to calculate, for each mesh size N_g, the mean error that the MFF makes on the GP predicted forces. This error is observed to diminish with a power law as a function of N_g. A negligible accuracy loss with the respect to the original GP model is achieved for $N_g \sim 10^6$ corresponding to about 100 grid points for each spline dimension. Since the potential was saved as a function of three distances, all going from a minimum of 1.5Å to the maximum cutoff distance of 4.5Å, this gives a point



(a) (b)

Figure 6.2: Computational cost of evaluating the 3-body energy (Eq. (6.2)) as a function of the database size N and the number of atoms M located within the cutoff radius. Panel (a): time taken for a single energy prediction using the GP (red solid line) and the mapped potential (blue dashed line), as a function of M, for a training set of $N = 5$ configurations. Panel (b): scaling of the same quantities as a function of N, for $M = 24$.

The system considered for this test is amorphous silicon.

to point spacing of 0.03Å.

Depending on the specific reference implementation, the speedup in calculating the local energy (Eq. (6.2)) provided by the mapping procedure can vary widely. However, it will always grow linearly with N and quadratically with M (see Figure 6.2), and it will always be substantial: in typical testing scenarios we found this to be of the order of 10^3–10^4. Obviously this huge speedup in the evaluation time of the force field comes at the cost of an increased memory usage since the spline interpolation points need to be stored, but for modern computer architectures this does not give rise to any practical concern.

Note that, while the procedure detailed here is based on the existence of a previously trained GP model, one might envision to skip this intermediate step and directly learn a *parametric* 3-body model on a grid points similarly arranged. This alternative approach is surely an attractive option which is worth exploring further, but the following difficulties should be kept in mind. Firstly, from the tests shown in Figure 6.1 it was found that with a local basis like the three dimensional spline used here the number of points needed to match the accuracy of the nonparametric GP is of the order of 10^5. Any linear model having that number of parameter would be very heavy to train (cubic

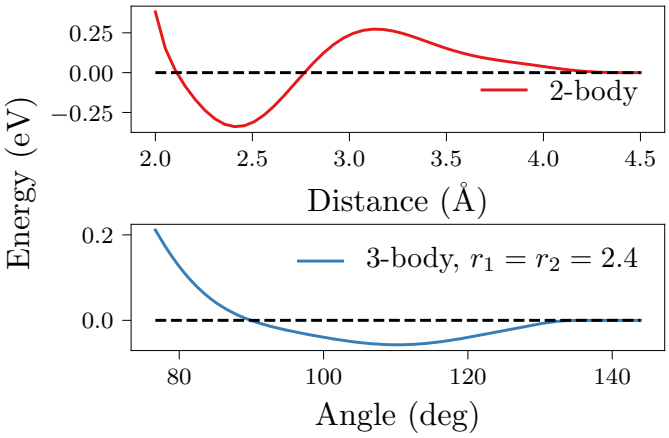


Figure 6.3: Energy profiles of the 2- and 3-body MFF, trained for the a-Si system at 650K. Upper panel: 2-body interaction term. Lower panel: 3-body interaction energy for an atomic triplet, angular dependence when the two distances from the central atoms are both equal to 2.4Å.

cost in the number of parameters) and even in the prediction phase (linear cost). Secondly, while it is likely that an appropriately chosen basis could potentially circumvent the first problem by requiring many fewer grid points, the problem of finding such a basis is not a trivial one and would need to be addressed. The MFF approach presented here, on the other hand, does not need to be based on any given linear expansion. Furthermore, the current spline implementation is particularly advantageous since the locality of the interpolation guarantees an extremely fast evaluation as only the 64 points closest to a given triplet need to be taken into account for any energy or force calculation.

Figure 6.3 shows the MFF obtained for a database of DFTB amorphous Silicon. As the 3-body kernel used for this test also included 2-body contributions (being the sum of the two kernels in Eqs. (3.14) and (3.14)), the corresponding MFF includes both types of interaction. As the energy profile is not prescribed by any particular functional form, it is free to optimally adapt to the information contained in the QM training set, to best reproduce the quantum interactions that produced it. The potential contains some expected features as e.g., a radial minimum at about $r \simeq 2.4$Å in the 2-body section (upper panel), the corresponding angular minimum at $\theta_0 \simeq 110°$ (lower panel), which is approximately equal to the sp^3 hybridization angle of 109.47°, and

rapid growth for small radii (upper panel) and angles (lower panel). Less intuitive features are also visible, which however contribute to the best representation of the bulk system's interactions that a 3-body expansion can achieve for the given database. An example is the shallow maximum in the 2-body section at $r \simeq 3.1$Å, which would of course disappear if we fitted our model on QM forces calculated for a Si dimer, that do not contain a hump. The resulting Si force field, appropriate for a Si dimer, would however inevitably reproduce the QM bulk interactions less accurately. More generally, training on the aggregate dataset could be a sensible compromise, producing a more transferable, but locally less accurate force field. Alternatively, an efficient strategy for simulating complex systems with space- and time-varying bonding nature might involve using (concurrently across the system, and at any given time) the locally optimal choice of low-order MFF, similarly to "mixture of experts" strategy suggested in Section 5.4 of the last chapter.

6.4 Summary

This chapter presented a simple and very effective method to substantially speed up the evaluation time of GP models using low order (practically 2- or 3-body) kernels. The procedure is made possible by first revealing the n-body nature of the GP. Once this is done, it is natural to recognise the n-body GP predictions and store them on a grid of points of a $3n-6$ dimensional space (one dimensional for $n=2$) corresponding to the effective degrees of freedom of n atoms. Subsequent calls to the force field can be then calculated by a local interpolation of the stored grid points (e.g., using a cubic spline). GP predictions computed this way are identical to the original ones but are much faster as they do not involve lengthy sums over database entries or expensive kernel evaluations.

The described method was here named "mapping" (and the resulting force field was named "MFF"), as one can imagine GP predictions to be mapped onto the effective degrees of freedom of n atoms. Building MFFs using simple regular grids becomes computationally impractical for $n > 3$ as the number of degrees of freedom quickly grows with the number n of atoms considered. Fortunately, many systems can be expected to be described very well by flexible 3-body force fields, and in this case the speedup achieved by building MFFs is

very substantial. To be more precise, the speedup in this case was predicted (and observed) to be linear with number of training configurations N and quadratic in the number of configuration atoms M. Within the reference implementation used and assuming typical values of N and M, MFFs were found to be faster than the corresponding GPs by factors of $10^3 - 10^4$, reaching the speed of very fast standard parametrised force fields. Differently from parametrised force fields, however, MFFs can be considered nonparametric and they can flexibly adapt to the shape that best reproduces the quantum calculations they are trained on. A particular MFF trained on amorphous silicon was shown, and its inspection revealed expected as well as unexpected features, both enhancing the final accuracy of the model. MFFs could be further improved by mapping the variance of the GP (providing this way a measure of uncertainty associated to their predictions), as well as by ensemble learning techniques consisting e.g., of an array of trained MFF each specialised on a given material phase.

Chapter 7

Gaussian process wave functions

7.1 Introduction

This chapter takes a step back away from the problems discussed in the rest of the thesis as it does not deal with the GP modelling of energies and forces. It can hence be read without a detailed knowledge of the other chapters, with the exception of Chapter 2, which contains the essential background material on GPs. Gaussian processes are here utilised to model the many-body wave functions of electrons. Electrons are fully quantum particles and, differently from classical ones, their state at any given time is described by a linear combination of all possible configurations available to the system. The total number of configurations grows exponentially with number of particles and this poses a great challenge in modelling quantum system. This challenge is often called the "quantum many-body problem", and the present chapter explores the possibility of facing it by representing the state of a quantum system compactly with a Gaussian process.

Sections 7.2 contains the necessary background material on quantum many-body physics and some of the methods available to tackle it. To ease the read, the information provided is very minimal and the reader is directed to external references for a more comprehensive exposition [106, 107]. The Hubbard model is first described, a prototypical model of strongly interacting electrons. Later, the Variational Monte Carlo (VMC) method is outlined, a standard modelling approach to quantum many-body systems based on the optimisation of a parametric Ansatz for the target wave function.

The Slater-Jastrow wave function is a particularly relevant Ansatz and

it is also briefly discussed. Finally, the an Ansatz based on a log-GP prior is proposed in Section 7.3, along with a range of specifically designed kernel functions. These are tested for the Hubbard model in Section 7.4. The results are promising but many possible improvements can and should be pursued, some of them are listed in Section 7.5.

7.2 Hubbard model and Variational Monte Carlo

The Hubbard Hamiltonian

Given a Hamiltonian operator $\hat{H}$, the lowest energy state of the system $|\phi_0\rangle$ can be found by solving the time independent Schrödinger equation

$$\hat{H}|\phi_i\rangle = E_i|\phi_i\rangle, \tag{7.1}$$

and selecting the lowest energy eigenstate.

For a wide range of many-body Hamiltonians, it is convenient to express the above eigenproblem in the *occupation number basis*, in which many-body basis vectors are given by

$$|\mathbf{n}\rangle = |n_1 n_2 \dots n_\alpha \dots \rangle, \tag{7.2}$$

where n_α is the number of particles populating a given single particle state α. In this chapter we will deal exclusively with systems of electrons, for which the occupation numbers n_α can only be 0 or 1 by the Pauli exclusion principle. The *creation operator* $c_\alpha^\dagger$ can be defined by its action on a generic many-body states in the occupation number basis as follows

$$c_\alpha^\dagger |n_1 \dots n_\alpha \dots \rangle = (-1)^{\sum_{j<\alpha} n_j} (1 - n_\alpha) \, |n_1, \dots, 1_\alpha, \dots \rangle. \tag{7.3}$$

The operator $c_\alpha^\dagger$ hence *creates* an electron in the single particle state α if $n_\alpha = 0$ (i.e., if the state is empty). On the other hand, if α is already populated ($n_\alpha = 1$), the action of $c_\alpha^\dagger$ annihilates the full many-body state, again as a consequence of the Pauli exclusion principle. Repeated application of the *creation operator* $c_\alpha^\dagger$ on a vacuum state $|0\rangle$ (where all single particle states are

empty) generates all many-body states as

$$|n_1 \dots n_\alpha \dots \rangle = \prod_\alpha (c_\alpha^\dagger)^{n_\alpha} |0\rangle. \tag{7.4}$$

Note that the phase factor $(-1)^{\sum_{j<\alpha} n_j}$ appearing in the definition of the creation operator (Eq. (7.3)) guarantees the anti-symmetry of the many-body states generated.

The *annihilation operator* is the adjoint of the creation operator $c_\alpha = (c_\alpha^\dagger)^\dagger$, and its action on an arbitrary state is given by

$$c_\alpha |n_1 \dots n_\alpha \dots \rangle = (-1)^{\sum_{j<\alpha} n_j} \, n_\alpha \, |n_1, \dots, 0_\alpha, \dots \rangle. \tag{7.5}$$

The annihilation operator removes one electrons from the single particle state α if α is populated, and annihilates the full many-body state if α is empty. From the above definitions one can show that creation and annihilation operators respect the following commutation relations

$$\{c_\alpha^\dagger, c_{\alpha'}^\dagger\} = 0, \; \{c_\alpha, c_{\alpha'}\} = 0, \; \{c_\alpha, c_{\alpha'}^\dagger\} = \delta_{\alpha\alpha'}, \tag{7.6}$$

where $\{a, b\}$ is the anticommutator between two operators $\{a, b\} = ab + ba$. The antisymmetry of the the fermionic basis states can be seen to follow directly from these relations. For instance, one can immediately show using (7.6) that a state with two electrons will change sign if the electrons are exchanged $c_\alpha^\dagger c_{\alpha'}^\dagger |0\rangle = -1 c_{\alpha'}^\dagger c_\alpha^\dagger |0\rangle$. To write down the Hubbard Hamiltonian, we need to define also the *number operator* $\hat{n}_\alpha = c_\alpha^\dagger c_\alpha$, which simply counts the number of electrons in state α.

For the one dimensional Hubbard model the single particle states can be conveniently indexed as $\alpha = (i, s)$, where i is the lattice site ($i = \{1, \dots, L\}$) and s is a spin variable ($s \in \{\uparrow, \downarrow\}$). In the occupation number basis just described, the Hubbard Hamiltonian reads [108, 109]

$$\hat{H} = -t \sum_{i=1}^{L} \sum_s (c_{i,s}^\dagger c_{i+1,s} + c_{i,s}^\dagger c_{i-1,s}) + U \sum_{i=1}^{L} \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} \,, \tag{7.7}$$

where L is the length of lattice and the indices of the operators are intended to have modulo L for periodic boundary conditions to apply.

The first term of the Hamiltonian, characterised by the parameter $t > 0$, is representative of the electrons' kinetic energy and, taken on its own, defines a non-interacting tight binding model where the hopping of an electron to a neighbouring site is favoured by a factor t. This term favours the delocalisation of electrons on the lattice. The second term is instead representative of the Coulomb repulsion between two electrons. This is modelled in a very essential way by means of an on-site repulsion of strength $U > 0$, which disfavours the movements of electrons on the lattice.

Both terms, taken on their own, lead to simple and analytically solvable Hamiltonians, while their interplay produces a complex Hamiltonian that is difficult to treat analytically and that is able to capture the essence a vast range of physical phenomena (as e.g., the Mott transition [110]).

The Hubbard model can be treated analytically in one dimension by means of the Bethe Ansatz [109], and this exact result will be very useful for benchmarking the proposed methodology. The physics of the Hubbard model in higher dimensions is instead still not entirely understood and it represents an active topic of research [111–113].

Variational Monte Carlo

A generic state for the one dimensional Hubbard model can be defined as a linear combination of all possible state vectors $|\mathbf{n}\rangle = |n_{1\uparrow} n_{1\downarrow} n_{2\uparrow} n_{2\downarrow} \dots n_{L\uparrow} n_{L\downarrow}\rangle$. For later notational convenience it is useful to define x_i to be the tuple $x_i = (n_{i\uparrow}, n_{i\downarrow})$, which allows the bases to be equivalently written as $|\mathbf{x}\rangle = |x_1 x_2 \dots x_L\rangle$. It is simple to see that for a Hubbard system of L sites and with $N_e^{\uparrow}$ ($N_e^{\downarrow}$) spin up (down) electrons there are $\binom{L}{N_e^{\uparrow}}\binom{L}{N_e^{\downarrow}}$ possible basis states $\{\mathbf{x}\}$ and a generic state can be written as

$$|\psi\rangle = \sum_{\{\mathbf{x}\}} \psi(\mathbf{x})|\mathbf{x}\rangle, \tag{7.8}$$

where the expansion coefficient $\psi(\mathbf{x})$ will be here called the *wave function* of the state.

It is a basic principle of quantum mechanics that the energy of any state $E = \langle\psi|\hat{H}|\psi\rangle$ cannot be lower than the ground state energy of the Hamiltonian E_0, and it can only be equal to it if the state is the ground state $|\psi\rangle = |\phi_0\rangle$ [114]. This is called the *variational principle* of quantum mechanics and it allows the calculation of the energy of a system using the so called *variational*

method summarised in the following. First, a suitable Ansatz is chosen for the given system, typically taking the form of a wave function $\psi^{\boldsymbol{\eta}}(\mathbf{x})$ providing the expansion coefficient for each term in Eq. (7.8) and depending only on the few parameters contained in the vector $\boldsymbol{\eta}$. These parameters are then optimised by minimising the corresponding energy. The process can in principle be repeated for different choices of the Ansatz, and the best Ansatz can be chosen as the one giving rise to the lowest ground state energy.

Optimising the energy would be impossible if that had to be calculated exactly by summing over all states as

$$\begin{aligned} E &= \sum_{\{\mathbf{x}\}\{\mathbf{x}'\}} \psi^*(\mathbf{x})\psi(\mathbf{x}')\langle\mathbf{x}|\hat{H}|\mathbf{x}'\rangle \\ &= \sum_{\{\mathbf{x}\}\{\mathbf{x}'\}} \psi^*(\mathbf{x})\hat{H}_{\mathbf{x}\mathbf{x}'}\psi(\mathbf{x}'). \end{aligned} \tag{7.9}$$

Luckily, the total energy of the Hubbard Hamiltonian can be efficiently calculated by Monte Carlo sampling. This can be done since the above expression can be rewritten as a classical average over the probability distribution given by the square of the wave function amplitude $|\psi(\mathbf{x})|^2$

$$\begin{aligned} E &= \sum_{\{\mathbf{x}\}\{\mathbf{x}'\}} \psi^*(\mathbf{x})\hat{H}_{\mathbf{x}\mathbf{x}'}\psi(\mathbf{x}') \\ &= \sum_{\{\mathbf{x}\}\{\mathbf{x}'\}} \psi^*(\mathbf{x})\frac{\psi(\mathbf{x})}{\psi(\mathbf{x})}\hat{H}_{\mathbf{x}\mathbf{x}'}\psi(\mathbf{x}') \\ &= \sum_{\{\mathbf{x}\}} |\psi(\mathbf{x})|^2 \sum_{\{\mathbf{x}'\}} \frac{\hat{H}_{\mathbf{x}\mathbf{x}'}\psi(\mathbf{x}')}{\psi(\mathbf{x})} \\ &= \langle E_L(\mathbf{x})\rangle_{|\psi(\mathbf{x})|^2}. \end{aligned} \tag{7.10}$$

In the last step of the above equation the *local energy* has been defined as $E_L(\mathbf{x}) = \sum_{\{\mathbf{x}'\}} \frac{\hat{H}_{\mathbf{x}\mathbf{x}'}\psi(\mathbf{x}')}{\psi(\mathbf{x})}$ and it is immediately obvious that the total energy can be computed as a classical average of this quantity. The local energy $E_L(\mathbf{x})$ for a given configuration $\mathbf{x}$ can be calculated in time linear in the size of the system. This can be seen from the fact that, although $E_L(\mathbf{x})$ formally involves a sum over the entire set $\{\mathbf{x}'\}$, only a linear number of terms will be non-zero as $\hat{H}_{\mathbf{x}\mathbf{x}'}$ will vanish everywhere else. A Markov chain [115] can easily be set to efficiently sample the probability function $|\psi(\mathbf{x})|^2$. In Variational Monte Carlo

[35] the total energy and its gradient are calculated as given above, and this allows the optimisation of a parametric wave function $\psi^{\boldsymbol{\eta}}(\mathbf{x})$.

Slater-Jastrow wave function

The Slater-Jastrow wave function is one of the earliest and most used wave function Ansatzes [39, 116–118]. It is based on the idea that a simple but nontrivial approximation to the exact ground state wave function can be improved by explicitly accounting for important and otherwise neglected correlations. The Slater-Jastrow wave function corrects the *Slater determinant* $\psi_S(\mathbf{x})$ providing the solution to a simpler Hamiltonian quadratic in the fermionic operators (such as e.g., the one obtained via a Hartree-Fock approximation), through an exponential function designed to capture specific many-body effects (the *Jastrow factor*). This gives

$$\psi_{SJ}(\mathbf{x}) = \mathrm{e}^{\lambda_{\boldsymbol{\eta}}(\mathbf{x})}\psi_S(\mathbf{x}), \tag{7.11}$$

where the parameters in $\boldsymbol{\eta}$ are chosen in order to minimise the energy.

7.3 Gaussian process wave functions

A log GP Ansatz

Here we try to improve on the Slater-Jastrow Ansatz by modelling the correction factor $\lambda_{\boldsymbol{\eta}}$ not with a specific parametric function but via a flexible Gaussian process. Equivalently, we assume the correction factor to be distributed according to a log Gaussian process:

$$\begin{aligned} \psi(\mathbf{x}) &= \mathrm{e}^{\lambda(\mathbf{x})}\psi_S(\mathbf{x}) \\ \lambda(\mathbf{x}) &\sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')). \end{aligned} \tag{7.12}$$

In such a way, contrary to standard VMC Ansatzes, the modelled wave function will not depend on a set of parameters $\boldsymbol{\eta}$ but directly on a set of training configurations and wave function amplitudes.

The model expressed in Eq. (7.12), from now on defined by a "Gaussian process wave function" (GP-WF), can be trained on a database of reference

configurations and wave function amplitudes $\mathcal{D} = \{(\mathbf{x}_d, \psi_d^r)\}_{d=1}^N$, and the predictions coming from the trained model take the form:

$$\begin{aligned} \hat{\psi}(\mathbf{x}) &= \mathrm{e}^{\hat{\lambda}(\mathbf{x})}\, \psi_S(\mathbf{x}) \\ \hat{\lambda}(\mathbf{x}) &= \sum_d k(\mathbf{x}, \mathbf{x}_d)\alpha_d \\ \alpha_d &= \sum_{d'} (\mathbf{K} + \mathbf{I}\sigma_n^2)^{-1}_{dd'} \log \frac{\psi^r_{d'}}{\psi_S(\mathbf{x}_{d'})}. \end{aligned} \tag{7.13}$$

The predicted wave function $\hat{\psi}$ looks very similar to the Slater-Jastrow parametric Ansatz in Eq. (7.11), with the key difference that the parametric correction $\lambda_{\boldsymbol{\eta}}$ in the exponential function has been substituted by a GP prediction. The reason why a log-GP Ansatz is particularly suited for our aim will be clear later. For the moment, it is sufficient to notice that if the GP predictions decompose *linearly* into contributions coming from groups of sites, then the wave functions will decompose *multiplicatively* in the same groups. This will in turn endow the Ansatz with good extrapolation properties, explored later on in this chapter.

The GP-WF proposed has the strong advantage of allowing the modelling of a much larger set of many-body correlations, since the computational cost of evaluating the wave function will not depend on their number. In fact, as detailed next a proper design of the kernel function allows modelling all possible many-body effects occurring in the system in polynomial time.

Kernel functions

Plaquette kernels In order to obtain a multiplicatively separable wave function, we want to model the log wave function $\lambda(\mathbf{x})$ in Eq. (7.12) in a linearly separable form. The simplest decomposition of this kind would involve writing $\lambda(\mathbf{x})$ as a linear sum of L contributions, each depending only on a single site, in a translational invariant manner:

$$\lambda_1(\mathbf{x}) = \sum_i \tilde{\lambda}_1(x_i), \tag{7.14}$$

where the function $\tilde{\lambda}(x_i)$ in the above is a GP that depends on the state of a single site only. The kernel function corresponding to this one-site decompo-

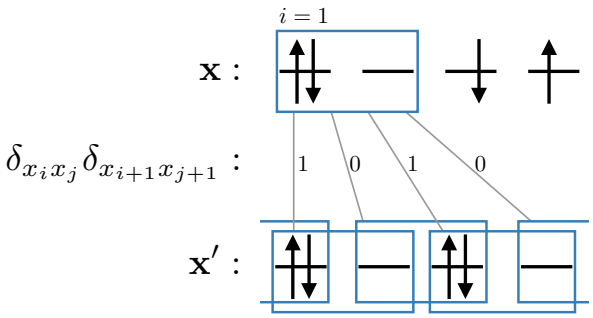


Figure 7.1: Illustration of the operations involved in the calculation of the n-site plaquette kernel (Eq. (7.17)). Plaquettes of a given size (in this case $n = 2$) are compared in the two configurations, and the kernel keeps track of the number of times two identical plaquettes are found. It is easy to generalise the reasoning in two or three dimensions since plaquettes can be given an arbitrary shape.

sition is trivially obtained as

$$\begin{aligned} k_1(\mathbf{x}, \mathbf{x}') &= \sum_{ij} \langle \tilde{\lambda}_1(x_i) \tilde{\lambda}_1(x'_i) \rangle \\ &= \sum_{ij} \delta_{x_i x'_j}, \end{aligned} \tag{7.15}$$

where the choice of a delta correlation $\langle \tilde{\lambda}_1(x_i) \tilde{\lambda}_1(x'_i) \rangle = \delta_{x_i x'_j}$ is very sensible in the discrete space considered here. The above reasoning can be generalised to model interactions involving an arbitrary number n of nearest neighbour sites. The decomposition of the log wave function in this case takes the form

$$\lambda_n(\mathbf{x}) = \sum_i \tilde{\lambda}_n(x_i, x_{i+1}, \ldots, x_{i+n-1}), \tag{7.16}$$

and a the corresponding n-site kernel is simply written as

$$k_n(\mathbf{x}, \mathbf{x}') = \sum_{ij} \delta_{x_i x'_j} \delta_{x_{i+1} x'_{j+1}} \ldots \delta_{x_{i+n-1} x'_{j+n-1}} . \tag{7.17}$$

The n-sites kernels can be understood—and extended to higher spatial dimensions—in terms of plaquettes. The kernel in Eq. (7.17) can indeed be imagined to scan through the configurations $\mathbf{x}$ and $\mathbf{x}'$ with a "plaquette" of

size n and a given shape (a line in one dimension) counting the number of times identical plaquettes are found. This procedure is illustrated in Figure 7.1 for a plaquette kernel of size $n = 2$.

Distance dependent kernels It is apparent that an n-site kernel with $n = L$ would be able to represent any correlation occurring in the system. Such a description would not be, however, a very useful one as the resulting GP will simply look for the coefficient with matching configuration in the database and predict that, with no hope of interpolation or extrapolation. A better model should allow for the possibility of selecting physically relevant many-body correlation effects. For instance, the interaction of two sites at a given distance might be more important than that of four consecutive sites. This objective could be achieved by relaxing the nearest neighbour constraint in the decomposition (7.16) and choosing plaquettes of arbitrary, physically based shapes. Albeit interesting and worth exploring, this approach suffers from the obvious drawback of having to choose "by hand" the relevant interactions modelled. A different route is here proposed.

To introduce a distance dependence, the following 2-site decomposition for the log coefficient can be envisioned

$$\lambda_2^d(\mathbf{x}) = \sum_{i_1 i_2} \tilde{\lambda}_2^d(x_{i_1}, x_{i_2}, \Delta_{i_1 i_2}) \tag{7.18}$$

where $\Delta_{i_1 i_2}$ is the relative position (signed distance in one dimension) of site i_2 with respect to i_1. Using a standard squared exponential kernel to learn functions on this distance, the kernel corresponding to the above decomposition can be written down as

$$k_2^d(\mathbf{x}, \mathbf{x}') = \sum_{i_1 i_2 j_1 j_2} \delta_{x_{i_1} x'_{j_1}} \delta_{x_{i_2} x'_{j_2}} \mathrm{e}^{-(\Delta_{i_1 i_2} - \Delta_{j_1 j_2})^2 / 2\ell^2}. \tag{7.19}$$

The cost of evaluating the above kernel scales as L^4, and can quickly become very computationally expensive. However, this cost does not increase if we make n larger. For instance, the cost of the next term in the series k_3^d can be brought down from L^6 to L^4 by implicitly representing higher 3-site correla-

tions as the square of 2-sites ones

$$\begin{aligned} k_3^d(\mathbf{x},\mathbf{x}') &= \sum_{i_1 j_1} \delta_{x_{i_1} x'_{j_1}} \left(\sum_{i_2 j_2} \delta_{x_{i_2} x'_{j_2}} \sum_{i_3 j_3} \delta_{x_{i_3} x'_{j_3}} \mathrm{e}^{-(\Delta_{i_1 i_2} - \Delta_{j_1 j_2})^2/2\ell^2} \mathrm{e}^{-(\Delta_{i_1 i_3} - \Delta_{j_1 j_3})^2/2\ell^2} \right) \\ &= \sum_{i_1 j_1} \delta_{x_{i_1} x'_{j_1}} \left(\sum_{i_2 j_2} \delta_{x_{i_2} x'_{j_2}} \mathrm{e}^{-(\Delta_{i_1 i_2} - \Delta_{j_1 j_2})^2/2\ell^2} \right)^2 . \end{aligned} \tag{7.20}$$

The above kernel gives rise to the decomposition

$$\lambda_3^d(\mathbf{x}) = \sum_{i_1 i_2 i_3} \tilde{\lambda}_3^d(x_{i_1}, x_{i_2}, x_{i_3}, \Delta_{i_1 i_2}, \Delta_{i_1 i_3}) \tag{7.21}$$

and, since $\Delta_{i_1 i_2}$ and $\Delta_{i_1 i_3}$ are *signed* distances, the function $\tilde{\lambda}_3^d$ is a unique function of the states of three sites and their relative position. It is now an immediate step to define a generic n-site distance dependent kernel as

$$k_n^d(\mathbf{x},\mathbf{x}') = \sum_{i_1 j_1} \delta_{x_{i_1} x'_{j_1}} \left(\sum_{i_2 j_2} \delta_{x_{i_2} x'_{j_2}} \mathrm{e}^{-(\Delta_{i_1 i_2} - \Delta_{j_1 j_2})^2/2\ell^2} \right)^{n-1} . \tag{7.22}$$

The L^4 scaling of the evaluation cost of distance dependent kernels can be reduced by exploiting the discontinuous nature of the relative positions Δ_{ij} in the lattice. Indeed, for a discontinuous variable one can safely use a delta correlation instead of a squared exponential one, which in turn allows a reduction of the computational cost to L^3. This can be seen by writing

$$\begin{aligned} k_n^d(\mathbf{x},\mathbf{x}') &= \sum_{i_1 j_1} \delta_{x_{i_1} x'_{j_1}} \left(\sum_{i_2 j_2} \delta_{x_{i_2} x'_{j_2}} \delta_{\Delta_{i_1 i_2} \Delta_{j_1 j_2}} \right)^{n-1} \\ &= \sum_{i_1 j_1} \delta_{x_{i_1} x'_{j_1}} \left(\sum_{\Delta} \delta_{x_{i_1+\Delta} x'_{j_1+\Delta}} \right)^{n-1} , \end{aligned} \tag{7.23}$$

where from the first to the second line the delta function constraint over the relative position is enforced by summing directly over the displacements.

Typically VMC runs only require the calculation of kernel variations from, say, $k(\mathbf{x},\mathbf{x}^r)$ to $k(\mathbf{x},\mathbf{x}^n)$, where the "new" configuration $\mathbf{x}^n$ is in the neighbourhood of the "root" configuration $\mathbf{x}^r$ (with the two being connected by the

action of the Hamiltonian). This adjustment can be computed in order L^2 operations as detailed in Appendix A.12.

Complete kernel A "complete" kernel, modelling the correlation of any number of sites at any distance, can at this point be defined as

$$k_c(\mathbf{x}, \mathbf{x}') = \sum_{i_1 j_1} \delta_{x_{i_1} x'_{j_1}} \left(1 + \mathrm{e}^{-1/\theta_o} \sum_{\Delta} \delta_{x_{i_1+\Delta} x'_{j_1+\Delta}} \mathrm{e}^{-\Delta^2/2\theta_d^2} \right)^{L-1} . \tag{7.24}$$

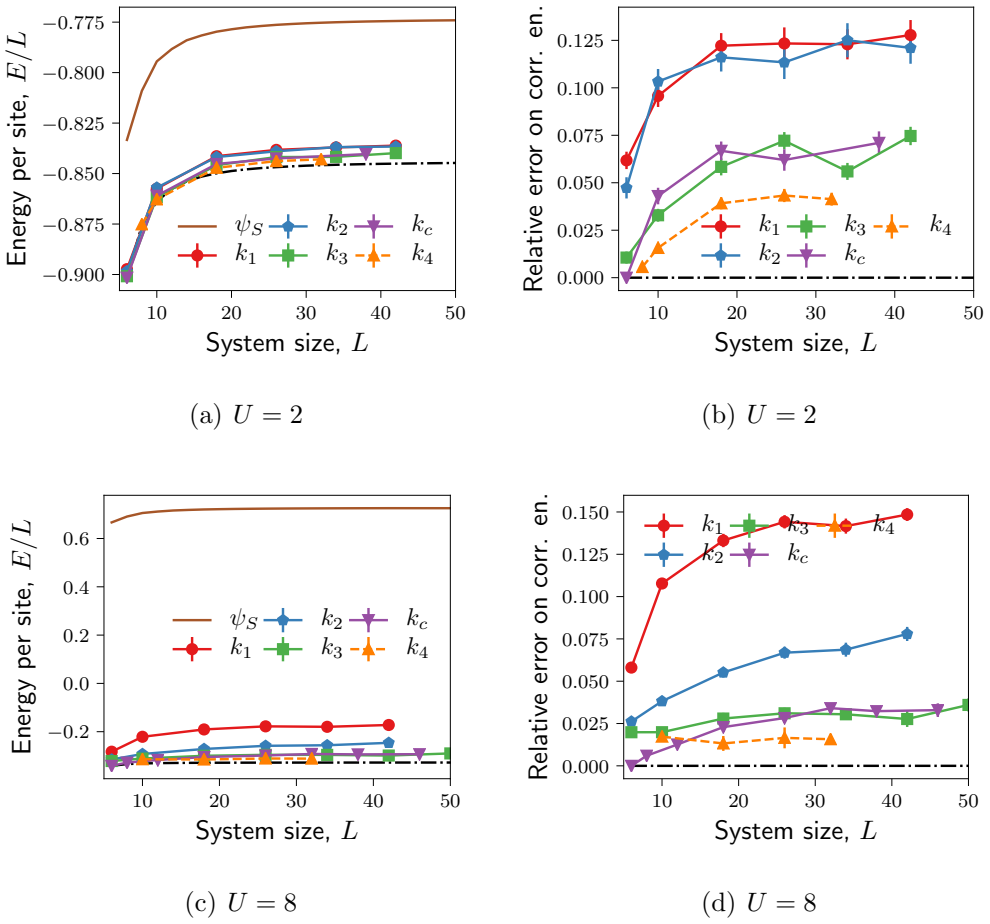
One can better visualise the effects by this kernel by imagining to expand the binomial in brackets. In fact, using the standard formula $(1+a)^p = \sum_{i=1}^{p} \binom{p}{i} a^i$ it is clear that all n-site distance dependent kernels k_n^d are present, with n going from one up to the full length L of the lattice. Correlations involving $i+1$ sites are super-exponentially suppressed by the factor $\binom{L-1}{i} \mathrm{e}^{-i/\theta_o}$, and the hyperparameter θ_o can be used to control this damping. Eq. (7.24) also includes a Gaussian dumping on the site to site distance Δ, controlled by the hyperparameter θ_d.

7.4 Tests on the Hubbard model

Testing size extensitivity

The kernels defined so far, combined with the log linear model of Eq. (7.12) have the property of inferring wave functions that are extensive with the lattice size. This means that in principle one could train a GP on a small system, for which exact results can be easily calculated, and use that to model a much larger—near thermodynamic—system. The amount of residual finite size effects can be imagined to depend on the particular choice of kernel, on its hyper-parameters, as well as on the size of the initial system on which the GP is trained on.

The following experiments were run to test on these ideas. The exact wave functions of 6-site and 8-site systems were first obtained by exact diagonalisation [119] and GP-WF models were fitted on them as given by Eq. (7.13). These trained model were then used to predict the energy of systems of up to 50 sites by the Monte Carlo sampling explained in Section 7.2. Since we look at one dimensional lattices, exact results are also available for benchmarking. In



(a) $U = 2$ (b) $U = 2$

(c) $U = 8$ (d) $U = 8$

Figure 7.2: The left column (panels (a) and(c)) reports the energy per site obtained by the different models as a function of L, the energy of the baseline Ansatz (ψ_S) and the that of DMRG (black, dashed-dotted line) are also shown. The right column (panels (b) and (d)) reports the relative errors on the correlation energy. All the models were fitted on 6-site data apart from the k_4 kernel (orange dashed line), which was trained on 8-site data.

particular, the thermodynamic energy can be calculated analytically using the Bethe Ansatz [109] while the energy for finite size systems can be computed in polynomial time using the Density matrix renormalisation group (DMRG) method [120–122].

Figure 7.2 reports the results of these experiments by graphing the achieved energy per site as a function of the size of the lattice, for $U = 2$ and for $U = 8$. All the tests presented here were run with $t = 1$, at half-filling, and with no net magnetisation. The figure shows that GP-WFs fitted on small systems

are able to represent surprisingly well the the wave function of larger systems as they can capture most of the correlation energy (defined as the difference between the mean field and the exact one). In particular, as clear from the first column (panels (a) and (c)) virtually all GP models very significantly improve on the baseline Ansatz ψ_S (solid brown curve). A more detailed picture can be captured from the second column (panels (b) and (d)), in which the relative error on the correlation energy is plotted.

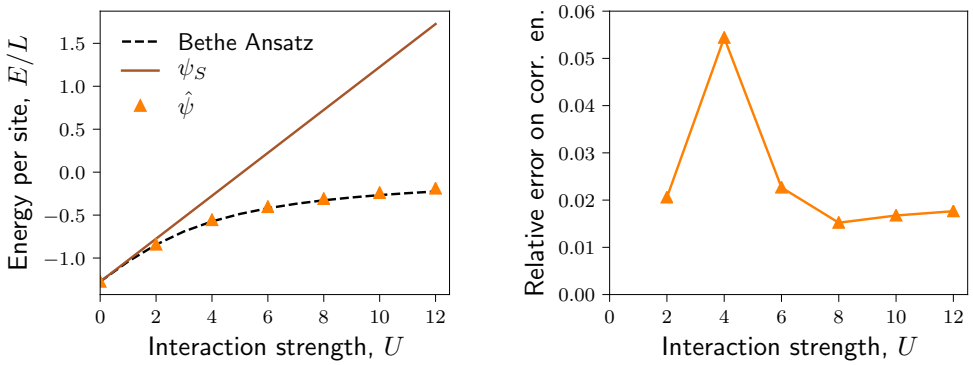
The following trends can be observed. Firstly, the error initially increases with system size before plateauing at some generally small value. As one would expect, the final error achieved is smaller for larger plaquette sizes when using simple k_n kernels. Perhaps surprisingly, the k_3 kernel trained on 6-site data does as well as the more complex complete kernel k_c for this system. This is probably due to the fact that only $L/2$ sites are completely explored in an L-site system at half filling (going from all empty to the all doubly occupied sites) so that not much is to be gained by going to larger plaquette sizes. The k_c has however the important property of being able of exactly reproducing the wave function on the original training data (zero error on the 6-site system), property that is absent in the k_n kernels shown. Finally, training on a larger initial systems yields a lower final energy as finite size effects in the data amplitudes (defining the correlated physics) are obviously less prominent.

The quality of the GP Ansatz with an n-site plaquette kernel is benchmarked in Figure 7.3 for several values of the interaction strength U. The left panel of the figure reports the energy per site of a 32-site system achieved by a 4-site kernel trained on the 8-site wave function, as a function of U. The exact thermodynamic energy and the baseline one are also reported. The right panel shows the percentage error on the correlation energy for the same models.

The comparison shows clearly that the proposed method achieves a high level of accuracy consistently across U.

Variational optimisation of database entries

The results shown so far are very promising and suggest that GP-based Ansatzes for wave functions could represent a new route to the description of strongly interacting electron system. In spite of this, there is an obvious pitfall in the approach presented so far that needs to be addressed: training on small sys-



(a) (b)

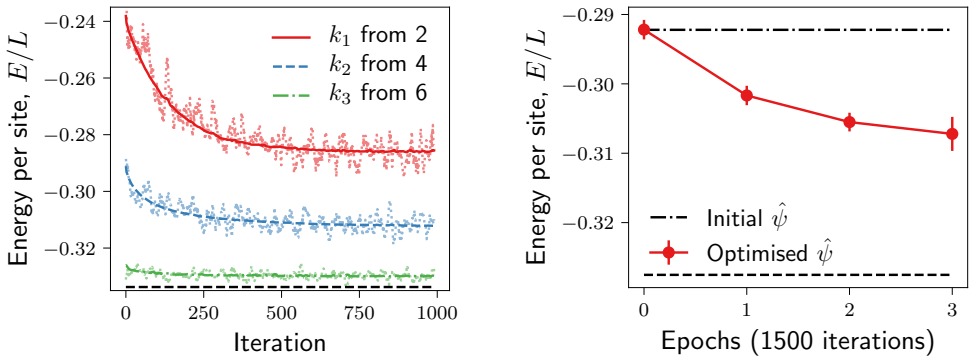
Figure 7.3: Precision of the GP-WF model as a function of U. The left plot shows the energy per site as given by the GP-WF model ($\hat{\psi}$), by the mean field baseline Staler determinant ψ_S, and by the Bethe Ansatz (dashed black line). The right plot presents the relative error on correlation energy achieved by the GP-WF model. The GP-WF model $\hat{\psi}$ was trained on 8-site data with a k_4 kernel and used to predict the energy of a 32-site system.

tems will always give rise to non-negligible finite size effects in the learned wave function.

The variational principle can help in tackling this problem. In particular, after training a GP-WF on a reference dataset $\mathcal{D} = \{(\mathbf{x}_i, \psi_i^r)\}_{i=1}^N$ belonging to a small system, the learned wave function could be optimised by adjusting the wave function entries belonging to training dataset $\{\psi_i^r\}_{i=1}^N$ in order to minimise the Monte Carlo sampled energy of the larger system studied. The simplest algorithm for performing this minimisation would be a steepest descent using the stochastic gradient obtained by Monte Carlo sampling (a stochastic gradient gradient descent [123]).

This idea is tested in Figure 7.4, which shows the energy per site of GP-WFs as a function of the iteration number of the gradient descent. In the left panel, Figure 7.4(a), a GP-WF is trained on a 2-, 4- or 6-site system and the learned wave function is optimised to minimise the energy of an 8-site system.

In the right panel, Figure 7.4(b), a 6-site GP-WF is instead optimised for a system of 32-sites. From the results shown, it is apparent that the proposed optimisation is able to significantly improve on the original fit. The improvement is however harder to achieve when larger training systems are employed. This is presumably a consequence of the larger number of database



(a) (b)

Figure 7.4: Energy per site as a function of gradient descent iteration. In the left panel the energy of an 8-site system is optimised starting from the wave function of a 2-, 4- and 6-site system, respectively using a k_1, k_2 and k_3 kernel respectively. In the right panel the energy of a 32-site system is optimised with respect to the database of a 6-site system using a k_3 kernel.

points that need to be optimised. In fact, database points here play the role of the variational parameters in standard VMC calculations and it is well known that VMC optimisations become very challenging with increasing number of parameters. Luckily, effective algorithms exist that can help in this direction [36, 123, 124] and their proper implementation for the present scope is crucial to fully explore the power of this approach.

7.5 Future extensions

The ideas and results presented in this chapter are very preliminary and many interesting directions of improvement can be identified. For instance, an obvious drawback of GP-WFs come from the cubic bottleneck in training a GP regression (cf. Eq. (7.13)). This is a well known issue in the ML community and several algorithms for "sparse GPs" have been proposed to mitigate it including e.g., Informative vector machines (IVMs) [125], the Relevance vector machines [126] or active learning approaches [127]. Preliminary tests have shown that IVMs can be trained successfully on wave functions of up to a million entries. However, while this is surely auspicious, a more thorough analysis and a careful comparison of the above listed methods is needed.

The variational optimisation could also be significantly improved. In fact, while here a simple stochastic gradient descent algorithm has been used, more sophisticated optimisation schemes (like stochastic reconfiguration [35, 36], the Adagrad or Adam algorithms [37, 123]) should be implemented and tested. Ideally, a more complete variational minimisation would also have to optimise the kernels hyperparameters. Another interesting route to explore would be expressing the GP Ansatz on a different basis as e.g., a momentum based representation.

Finally, it is fundamental to test the proposed framework on more challenging systems. The perfect candidate would be the two dimensional Hubbard model as many feature of this system remain not well understood. This system, differently from the one dimensional analogue, presents substantial modelling challenges [111]. Perhaps the major difficulty is the nontrivial sign structure of the wave function (absent in one dimension). The GP-WF approach presented here relies on the Slater determinant sign but this could be augmented or substituted by a separate machine learning algorithm specifically designed to learn the sign of a given configuration from data. Good candidates for this task could be kernel based algorithms like Gaussian process classification [47] or support vector machine classification [48]. Alternatively, sign information could be included in the present model by the use of complex Gaussian processes [128].

7.6 Summary

This chapter introduced a novel method to model wave functions of strongly interacting electronic systems. The method is centred on the idea of compactly representing the sought wave function not in terms of a few parameters—as typically done in VMC approaches—but in terms of a data set of calculations. A log-GP Ansatz was proposed and a range of kernel functions encompassing specific sets of physically based correlation effects was proposed. These kernels were then tested on the one dimensional Hubbard model and they were shown to represent surprisingly well the wave functions of large lattice systems ($\sim$ 40 sites) even when trained on a very small lattices ($\sim$ 6 sites). Obviously, finite size effects are always present in the learned wave function when only training on small systems. To overcome this limitation, a variational scheme

was proposed. This involves optimising entries of the training database in order to minimise the variational energy of the target system calculated via Monte Carlo sampling. The optimisation was shown to significantly reduce the finite size effects of the learned wave function, but the improvements were found to be harder to obtain as the training system size increases. This is a common problem of stochastic optimisations and it could be tackled by faster and more sophisticated routines and and a parallel implementation. The results presented here are very promising and make GP wave functions a potentially very efficient candidate for the description of quantum many-body systems. However, more research is surely necessary to fully explore their capability and some extension of the work presented here have been proposed.

Chapter 8

Conclusions

This thesis developed data-driven models for atomic force fields and for electronic wave functions, which were designed with the aim of being not only flexible and accurate interpolators but also fast to evaluate and easy to interpret. The models were constructed within the Gaussian process regression framework by a careful design of a set of GP kernel functions made to include as much prior information as is available on the functions to be learned.

From Chapter 2 to Chapter 6, the main focus of this thesis has been on the constructing data-driven models for atomic force fields that are flexible enough to achieve a satisfactory accuracy and simple enough to allow fast evaluation and good transferability. With this aim, a range of GP kernels was developed, all encoding the force field smoothness as well as its transformation properties upon rigid translations, rotations and reflections and upon permutations of atoms. These kernels were also designed to give rise to force field of a prescribed interaction order n, intimately related to the complexity of the model. Kernels with a finite n give rise to n-body force fields while kernels for which n is infinite give rise to fully many-body models.

The kernels developed for learning force fields can be further divided into two categories. The first one is that of scalar local energy kernels, learning local energy function that can then be differentiated to provide the sought force field. The second one is that of matrix-valued force kernels, which instead learn a force field directly without passing through an intermediate energy expression. Local energy kernels always give rise to conservative force fields, which can then be used for any molecular dynamics simulation, including those that require a constant energy. On the contrary, constant energy cannot in general be

achieved using the matrix-valued force kernels proposed here. However, the very high force accuracy required by mixed QM/MM approaches, along with the impossibility of conserving energy exactly in any online learning scheme, might often justify their use. Moreover, the force kernels defined here, as well as the general methodology developed for constructing and using them, can be exploited to learn any other physical quantity of interest that possesses the same symmetry properties. The GP models developed were shown to be accurate interpolators for a range of systems simulated at the DFT level of accuracy, outperforming the force accuracy of traditional parametrised force fields and being competitive with other machine learning force fields.

The variety of different models developed in this thesis (and elsewhere) makes the problem of selecting the single model best suited for a given system unavoidable. A principled way of performing this choice is given by the selection of the simplest possible model able to correctly reproduce the system's interaction. In practice, this corresponds to the lowest n model providing a target accuracy or to the model with the largest marginal likelihood. In both cases, it was found that more flexible (higher order) GP kernels are not, in general, optimal. On the contrary, low order kernels should be used to model not only relatively "simple" systems but also more complex systems when the training dataset is not large enough to fully resolve their interactions. This is a consequence of the fact that higher order models require more data to be properly trained and tend to generalise worse to new configurations not well represented in the training database. Low order models instead converge quickly to their final accuracy, which can be satisfactory or not depending on the application at hand, and can be expected to yield meaningful predictions also on unseen configurations, for which a more complex (higher order) model might badly extrapolate.

Using low order kernels was further shown to be very advantageous in terms of evaluation time. In fact, the predictions of a trained 2- or 3-body GP model can be sped up by orders of magnitude by calculating and storing the value of the learned potential energy on a grid of points within the space of the effective degrees of freedom of the n-plet (e.g., a set of distances). The resulting force field, named "mapped" force field (MFF), is extremely fast to evaluate (being as fast as standard parametrised potential) but it is also typically very accurate since its nonparametric character allows it to optimally adapt to reference

database calculations. Furthermore, MFFs can be generated automatically without the need of a fine-tuned parametrisation. For the above reasons an MFF could represent the model of choice for many challenging applications requiring an accurate force field that is also able to probe very long timescales or very large systems sizes [1][60].

In addition to constructing the learning models described above this thesis has also shed light on well known past approaches. In particular, it has clarified the use and need of the Haar-integration to generate symmetric kernels (e.g., in relation to the famous SOAP kernel [32]) and it has included the interaction order as an important feature of any ML-FF. More generally, it has created a clear and coherent theoretical framework for the development of data-driven models for force field learning, which lays the basis for many possible future research directions.

Three possible research directions are indicated in the following. Firstly, the scalability of the developed GP models can be dramatically improved. In fact, standard GP regression has a cubic time complexity and quadratic memory complexity in the number of database entries, limiting the practical use of GPs only to a few thousand configurations. This bottleneck can easily become a practical problem when training on large and heterogeneous databases, in which simply selecting a manageable subset at random will not provide satisfactory accuracy. When this is the case, sparse variants of GP regression should be implemented and tested. These can either take the form of a non-uniform and information efficient subsampling of the training database (as in the case of the Informative vector machines [125], Relevance vector machines [131, 132] and methods based on a CUR decomposition of the kernel matrix [133]), or involve a variational scheme for the optimisation of a small number of inducing points in order to represent a much larger database of calculations [134, 135]. A second line of research would involve exploring the potential of ensemble learning approaches to improve the accuracy of GP models for systems that undergo structural or chemical transformations. In such systems, one can envision different GP models (or even different MFFs) to specialise on a given type of environment, and a separately trained gating function to decide which model (or which linear combination of models) to use at any given time. The

[1] A Python implementation of the automatic construction of MFFs (to be used within an "ASE" environment [129]) from a reference database is freely available online [130].

division of the full database into similar regions could be done in a previous step by using algorithms of clustering or dimensionality reduction based on the distance provided by the chosen kernel function. Alternatively, the clustering step could be performed concomitantly with the training of the GPs and of the gating function, presumably yielding to better accuracy and to a larger computational cost [99]. Finally, the potential of some of the models proposed within on-line learning schemes could be explored. In such schemes, a trained GP model would be used for predictions during an MD run until a given error measure (e.g., the GP predicted standard deviation) is below a chosen threshold. If the error is found to be larger than this threshold, a DFT calculation could be performed and the model modified to take the new information into account.

In Chapter 7, this thesis has also explored the capabilities of a log-GP Ansatz to reproduce the many-body correlations of a wave function of a system of electrons. Different types of kernels have been designed and tested, each corresponding to a well defined set of correlations.

The particular structure of the log-GP model and of the kernels proposed imposes an exact product separability to the predicted wave function, which guarantees good extrapolative power across different system sizes. For this reason, the log-GP model can be trained on an exact wave function of a small system and its predictions can be effectively used to compute the properties of a larger system whose exact wave function cannot be calculated. However, residual finite size effects can always be expected to be present. These can be moderated exploiting the variational principle by optimising the database wave function in order to minimise the energy of the target system. The methodology was tested on the one dimensional Hubbard model, for which the exact wave function can be easily computed even on large system sizes. The promising initial results found indicate that the proposed model could represent a new route for the construction of compact wave functions for VMC calculations, based on a set of data points rather than on a set of parameters.

However, more research is surely needed to assess the potential of this approach. In particular, it would be interesting to benchmark the log-GP Ansatz on the two dimensional Hubbard model. In comparison to the one dimensional analogue, this possesses a range of possible geometries and many competing orders, ultimately giving rise to a much richer phase diagram which

is still subject of scientific debate and active research [111, 136].

From the perspective of extending the GP Ansatz developed here from the one dimensional to the two dimensional Hubbard model, we can identify two main challenges. On the one hand, the ground state wave function of the two dimensional model possesses a nontrivial sign structure absent in the one dimensional case. On the other hand, the system sizes needed to probe the thermodynamic properties of the two dimensional Hubbard model are much larger.

The first issue translates to the problem of predicting the sign of each wave function configuration, and this could be tackled in different ways. The simplest option would be relying on the sign predicted by the exact wave function of a solvable quadratic Hamiltonian which in a mean field sense models the phase probed in the system (as e.g., the Hartree-Fock Hamiltonian). Alternatively, the sign could be learned along with the amplitude by using a single complex-valued log-GP Ansatz. The phase predicted by this Ansatz would not, however, be constrained to be either 0 or π (as in the case of the exact wave function being modelled) but it could take values in the entire range of possible angles, and it is not clear how much this would affect the Ansatz performance. Finally, one might attempt to make use of a separate classification algorithm specifically designed to predict only the sign of a given configuration. Obvious candidates for this learning task would be kernel based algorithms like GP or SVM classification. However, it remains to be checked whether the kernels developed here would also be suited for accurate sign prediction.

The second challenge presented by the two dimensional Hubbard Hamiltonian i.e., the need to access larger system sizes, translates within the GP framework into a problem of scalability. As in the case of force field learning, possible solutions to the poor scalability of standard GP regression can be given by sparse GP approaches, which should also in this case be implemented and tested. However, differently from the GP models for force fields, GP wave functions also require a variational optimisation of the database entries. In this case, to address the scalability issue while also minimising the finite size effects of the Ansatz, one can envision a two step optimisation algorithm. In a first step, the entries already present in the database of the GP wave function are optimised to minimise the target system energy. In a second step, the database is enlarged by adding configurations from the target systems and it

is later sparsified in order to keep the number of entries manageable.

The continued iteration of this two steps should lead the algorithm to adaptively find the inducing configuration-amplitude pairs that optimally represent the sought wave function. Clearly, a long convergence time as well as an intractable size of the converged inducing database could make the algorithms impractical or might call for further research efforts. However, the preliminary results presented in this thesis suggest that this is a path worth pursuing.

Appendices

A.1 On the derivation of the GP predictive distribution

This appendix gives a sketch of the procedure by which Eq. (2.7) is obtained, which substantially relies on the properties of multivariate Gaussian distributions. For full details on this one can consult the excellent Refs. [48] and [47].

In the main text, the probability distribution $p(\boldsymbol{\varepsilon}^r \mid \boldsymbol{\rho})$ (i.e. the *marginal likelihood* of the data) was computed in closed form (Eq. (2.6)). To calculate the predictive distribution for the new pair (ρ^*, ε^*) (i.e. $p(\varepsilon^* \mid \rho^*, \boldsymbol{\varepsilon}, \boldsymbol{\rho})$), one can first write down the probability of the original dataset augmented with the new pair by adapting Eq. (2.6) to the augmented database:

$$\begin{cases} p(\varepsilon^*, \boldsymbol{\varepsilon}^r \mid \rho^*, \boldsymbol{\rho}) & = \mathcal{N}(\mathbf{0}, \mathbf{C}_a) \\ \mathbf{C}_a & = \begin{pmatrix} \mathbf{C} & \mathbf{k} \\ \mathbf{k}^{\mathrm{T}} & k(\rho^*, \rho^*) \end{pmatrix} . \end{cases}$$

The two variables ε^* and $\boldsymbol{\varepsilon}^r$ are hence distributed jointly according to a multivariate normal distribution. A consequence of this is that the conditional distribution of one variable conditioned on the other is also normal. The corresponding mean and variance of this conditional distribution $p(\varepsilon^* \mid \rho^*, \boldsymbol{\varepsilon}^r, \boldsymbol{\rho})$ can be found by simple "completion of the square" in the argument of the exponential of the joint distribution (see Ref. [48]). Doing so results in the predictive distribution in Eq. (2.7).

A.2 Proof of the optimality of the predictive mean

The predictive distribution $p(\varepsilon^* \mid \rho^*, \mathcal{D})$ in Eq. (2.7) ($\mathcal{D} = (\boldsymbol{\varepsilon}^r, \boldsymbol{\rho})$), completely specifies our knowledge about the local energy function ε^* associated to a given configuration ρ^*. Once such a configuration is encountered during an MD run one is typically faced with the problem of deciding a single estimate $\bar{\varepsilon}(\rho)$ for its energy in order to proceed to the subsequent time step.

This can be formally seen as decision making problem under conditions of uncertainty. A principled solution from decision theory [48, 137] consists in choosing the estimate which minimises a given expected loss under the known distribution

$$\langle L \rangle = \int d\varepsilon^* \, p(\varepsilon^* \mid \rho, \mathcal{D}) L(\bar{\varepsilon}(\rho), \varepsilon^*).$$

The loss function L can be taken to be, for instance, the squared error between the estimate and the measured value of the local energy $L_{SE} = (\bar{\varepsilon}(\rho) - \varepsilon^*)^2$. Minimising the expected squared error loss with respect to the function $\bar{\varepsilon}(\rho)$ can be done analytically by equating the relative functional derivative to zero

$$\begin{aligned} \frac{\delta \langle L_{SE} \rangle}{\delta \bar{\varepsilon}(\rho)} &= 2 \int d\varepsilon^* \, p(\varepsilon \mid \rho, \mathcal{D})(\bar{\varepsilon}(\rho) - \varepsilon^*) \\ &= 2(\bar{\varepsilon}(\rho) - \langle \varepsilon^* \rangle) = 0, \end{aligned}$$

where from the last line it is clear that the optimal estimate is the mean $\hat{\varepsilon}(\rho)$ of the predictive distribution $p(\varepsilon^* \mid \rho^*, \mathcal{D})$ in Eq. (2.7)(or, equivalently, the mean of the posterior GP). One can show that using the absolute error loss function $L_{AE} = |\bar{\varepsilon}(\rho) - \varepsilon|$ makes the mode of the predictive distribution the optimal estimate, coinciding with the mean in the case of Gaussian distributions.

A.3 Kernels for multiple chemical species

This appendix develops the basic theory to construct kernels for multispecies systems and provides specific expressions for the case of 2- and 3-body kernels. What presented is based on the concepts described in Chapters 2 and 3, reading such chapters before continuing with this appendix is hence recommended.

It is convenient to show the reasoning behind multispecies kernel construction starting from a simple example. Defining by s_j the chemical species of

atom j, a generic 2-body decomposition of the local energy of an atom i surrounded by the configuration ρ_i takes the form

$$\varepsilon(\rho_i) = \sum_{j \in \rho_i} \phi^{s_i s_j}(r_{ij}).$$

where a pairwise function $\phi^{s_i s_j}(r_{ij})$ is assumed to provide the energy of each couple of atoms i and j depending on their distance r_{ij} and on their chemical species s_i and s_j. These pairwise energy functions should be invariant upon re-indexing of the atoms i.e., $\phi^{s_i s_j}(r_{ij}) = \phi^{s_j s_i}(r_{ji})$. The kernel for the function $\varepsilon(\rho_i)$ then takes the form

$$\begin{aligned} k_2^s(\rho_i, \rho'_l) &= \langle \varepsilon(\rho_i)\varepsilon(\rho'_l) \rangle \\ &= \sum_{jm} \langle \phi^{s_i s_j}(r_{ij}) \phi^{s'_l s'_m}(r'_{lm}) \rangle \\ &= \sum_{jm} \tilde{k}^{s_i s_j s'_l s'_m}(r_{ij}, r'_{lm}) \end{aligned}$$

The design problem is at this point reduced to that of choosing a suitable kernel $\tilde{k}^{s_i s_j s'_l s'_m}$ comparing couples of atoms. An obvious choice for this would include a simple squared exponential for the radial dependence and a delta correlation for the species dependence, giving rise to $\delta_{s_i s'_l} \delta_{s_j s'_m} k_{SE}(r_{ij}, r'_{lm})$. This kernel is however still not symmetric upon the exchange of two atoms and it would hence not impose the required property $\phi^{s_i s_j}(r_{ij}) = \phi^{s_j s_i}(r_{ji})$ on the learned pairwise potential. Permutation invariance can be simply enforced by a direct sum over the permutation group, in this case simply an exchange of the two atoms l and m in the second configuration. The resulting 2-body multispecies kernel reads

$$k_2^s(\rho_i, \rho'_l) = \sum_{\substack{j \in \rho \\ m \in \rho'}} (\delta_{s_i s'_l} \delta_{s_j s'_m} + \delta_{s_i s'_m} \delta_{s_j s'_l}) \mathrm{e}^{-(r_{ij} - r'_{lm})^2 / 2\ell^2}$$

This can be considered the natural generalisation of the single species 2-body kernel in Eq. (3.14). A very similar sequence of steps can be followed for the 3-body case. By defining the vector containing the chemical species of an ordered triplet as $\mathbf{s}_{ijk} = (s_i s_j s_k)^{\mathrm{T}}$ as well as the vector containing the corresponding three distances $\mathbf{r}_{ijk} = (r_{ij} r_{jk} r_{ki})^{\mathrm{T}}$, a multispecies 3-body kernel

can be compactly written down as

$$k_3^s(\rho_i, \rho'_l) = \sum_{\substack{j>k\in\rho_i \\ m>n\in\rho'_l}} \sum_{\mathbf{P}\in\mathcal{P}} \delta_{\mathbf{s}_{ijk},\mathbf{P}\mathbf{s}'_{lmn}} \mathrm{e}^{-\|\mathbf{r}_{ijk}^{\mathrm{T}}-\mathbf{P}\mathbf{r}'_{lmn}\|^2/2\ell^2},$$

where the group $\mathcal{P}$ contains six permutations of three elements, represented by the matrices $\mathbf{P}$. The above can be considered the direct generalisation of the 3-body kernel in Eq. (3.15). It is simple to see how the reasoning can be extended to an arbitrary n-body kernel. I Importantly, the computational cost of evaluating the multispecies kernels described above does not increase with the number of species present in a given environment, and the kernels' interaction order could be increased arbitrarily at no extra computational cost using Eqs. (3.18) and (3.19).

A.4 Kernel order by explicit differentiation

To prove that the kernel given in Eq. (3.2) is 2-body in the sense of Eq. (2.28) it is sufficient to show that its second derivative with respect to the relative position of two different atoms of the configuration ρ always vanishes. The first derivative is

$$\begin{aligned}
\frac{\partial k_2(\rho,\rho')}{\partial \mathbf{r}_{i_1}} &= \sum_{ij} \frac{\partial}{\partial \mathbf{r}_{i_1}} \mathrm{e}^{-\|\mathbf{r}_i-\mathbf{r}'_j\|^2/2\ell^2} \\
&= \sum_{ij} \mathrm{e}^{-\|\mathbf{r}_i-\mathbf{r}'_j\|^2/2\ell^2} \frac{(\mathbf{r}_i-\mathbf{r}'_j)}{\ell^2}\delta_{ii_1} \\
&= \sum_{j} \mathrm{e}^{-\|\mathbf{r}_{i_1}-\mathbf{r}'_j\|^2/2\ell^2} \frac{(\mathbf{r}_{i_1}-\mathbf{r}'_j)}{\ell^2}.
\end{aligned}$$

This depends only on the atom located at $\mathbf{r}_{i_1}$ of the configuration ρ. Thus, differentiating with respect to the relative position $\mathbf{r}_{i_2}$ of any other atom of the configuration gives the relation in Eq. (2.28) for 2-body kernels:

$$\frac{\partial^2 k_2(\rho,\rho')}{\partial \mathbf{r}_{i_1}\partial \mathbf{r}_{i_2}} = 0.$$

It is now straightforward to also show that the kernel defined in Eq. (3.3) is n-body in the sense of Eq. (2.28). Indeed, this follows naturally from the

result above, given that k_n is defined as $k_n = k_2^{n-1}$. We can thus write down its first derivative as

$$\frac{\partial k_n}{\partial \mathbf{r}_{i_1}} = (n-1)k_2^{n-2}\frac{\partial k_2}{\partial \mathbf{r}_{i_1}}.$$

Since the second derivative of k_2 is null, the second derivative of k_n is simply

$$\frac{\partial^2 k_n}{\partial \mathbf{r}_{i_1}\partial \mathbf{r}_{i_2}} = (n-2)(n-1)k_2^{n-3}\frac{\partial k_2}{\partial \mathbf{r}_{i_1}}\frac{\partial k_2}{\partial \mathbf{r}_{i_2}}$$

and after $n-1$ derivations we similarly obtain

$$\frac{\partial^2 k_2^{n-1}}{\partial \mathbf{r}_{i_1}\cdots\partial \mathbf{r}_{i_n}} = (n-1)!\, k_2^0\,\frac{\partial k_2}{\partial \mathbf{r}_{i_1}}\cdots\frac{\partial k_2}{\partial \mathbf{r}_{i_{n-1}}}.$$

Since $k_2^0 = 1$, the final derivative with respect to the n_{th} particle position $\mathbf{r}_{i_n}$ is zero as required by Eq. (2.28).

A.5 A first one dimensional toy model

To test the ideas behind the n-body kernels, we used a one dimensional n'-particle model in reference system where a ("central") particle is kept fixed at the coordinate axis origin (consistent with the local configuration convention of Eq. (3.1)). The energy of the central particle is defined as

$$f = \sum_{i_1\ldots i_{n'-1}} J\, x_{i_1}\ldots x_{i_{n'-1}}$$

where $\{x_{i_p}\}_{p=1}^{n'-1}$ are the relative positions of $n'-1$ particles, and J is an interaction constant.

To generate Figure 3.1 a large set of configurations was generated by uniformly and independently sampling each relative position x_{i_p} within the range $(-0.5, 0.5)$. The energy of the central particle of each configuration was then given by the above equation, with the interaction constant J set to 0.5. In order to analyse the converged properties of the n-body kernels presented, large training sets ($N = 1000$) were used.

A.6 Databases details

Extra details on the quantum mechanical datasets used for the tests presented in this thesis are given below. Materials of four chemical species where considered, the corresponding radial cutoffs used to generate the local environments are: 4.45 Å (Ni), 4.45 Å (Fe), 3.7 Å (C) and 4.5 Å (Si).

Nickel and iron

The Ni and Fe databases were obtained through DFT calculations with the electronic exchange and correlation interactions modelled via the PBE/GGA approximation [85].

For crystalline Ni and Fe, simulations were performed using a $4 \times 4 \times 4$ periodically repeated unit cell, and controlling the temperature by means of a weakly-coupled Langevin thermostat (the DFT trajectories are available from the King's College London research data management system at the link `http://doi.org/10.18742/RDM01-92`).

The cluster Ni database was obtained simulating a Ni_{19} nanoparticle at constant temperature using a Nose-Hoover thermostat, with an initial geometry given by four stacked hcp layer.

Carbon

The C database comprises a variety of structures including bulk diamond, AB and ABC stacked graphene layers and amorphous structures. The DFT calculations were performed at different temperatures, pressures. The database is available in the "libAtoms" data repository via the following link `http://www.libatoms.org/Home/DataRepository`.

Silicon

Crystalline and amorphous Si databases were obtained from DFTB molecular dynamics simulations of 64 atoms in periodic boundary conditions. A Langevin thermostat was used to control the temperature in the crystalline system, while the amorphous system was evolved at constant energy.

A.7 Covariant integration of 2-body kernels

The integral we wish to evaluate for the covariant 2-body kernel in two or three spacial dimensions ($d \in \{2, 3\}$) is

$$\begin{aligned} \mathbf{K}_2^{SO(d)}(\rho, \rho') &= \sum_{ij} \int_{SO(d)} d\mathcal{R}\, \mathbf{R} \mathrm{e}^{-(\mathbf{r}_i - \mathbf{R}\mathbf{r}'_j)^2/2\ell^2} \\ &= \sum_{ij} I_{ij}. \end{aligned}$$

First of all it is convenient to separate the radial part from the angular one as the former does not depend on rotations:

$$\begin{aligned} I_{ij} &= \mathrm{e}^{-(r_i^2 + r_j'^2)/2\ell^2} \int d\mathcal{R}\, \mathbf{R}\, \mathrm{e}^{\mathbf{r}_i^{\mathrm{T}} \mathbf{R} \mathbf{r}'_j/\ell^2} \\ &= C_{ij} \int d\mathcal{R}\, \mathbf{R}\, \mathrm{e}^{\mathbf{r}_i^{\mathrm{T}} \mathbf{R} \mathbf{r}'_j/\ell^2}. \end{aligned}$$

2D systems

If we define $\mathbf{R}_{ij}$ to be the rotation matrix that brings the vector $\mathbf{r}'_j$ onto $\mathbf{r}_i$, then we can perform the change of variable $\tilde{\mathbf{R}} = \mathbf{R}\mathbf{R}_{ij}^{\mathrm{T}}$

$$\begin{aligned} I_{ij} &= C_{ij} \int d\tilde{\mathcal{R}}\, \tilde{\mathbf{R}}\, \mathrm{e}^{\mathbf{r}_i^{\mathrm{T}} \tilde{\mathbf{R}} \mathbf{R}_{ij} \mathbf{r}'_j/\ell^2} \mathbf{R}_{ij} \\ &= C_{ij} \int d\tilde{\mathcal{R}}\, \tilde{\mathbf{R}}\, \mathrm{e}^{\mathbf{r}_i^{\mathrm{T}} \tilde{\mathbf{R}} \tilde{\mathbf{r}}'_j/\ell^2} \mathbf{R}_{ij}. \end{aligned}$$

where the two vectors $\mathbf{r}_i$ and $\tilde{\mathbf{r}}_j$ are now aligned with each other. By parametrising all rotations by a single angle θ we can rewrite the above integration as

$$\begin{aligned} I_{ij} &= C_{ij} \int_0^{2\pi} \frac{d\theta}{2\pi}\, \mathbf{R}(\theta) \mathrm{e}^{\mathbf{r}_i^{\mathrm{T}} \mathbf{R}(\theta) \tilde{\mathbf{r}}'_j/\ell^2} \mathbf{R}_{ij} \\ &= C_{ij} \left(\int_0^{2\pi} \frac{d\theta}{2\pi}\, \mathbf{R}(\theta) \mathrm{e}^{r_i r'_j \cos\theta/\ell^2} \right) \mathbf{R}_{ij}. \end{aligned}$$

The integral in brackets can now be given an analytic form. The rotation matrix $\mathbf{R}(\theta)$ is composed by $\cos\theta$ on the diagonal and $\{\sin\theta, -\sin\theta\}$ off the

diagonal. Evaluating the above integration for such terms one finds that

$$\begin{cases} \int_0^{2\pi} \frac{d\theta}{2\pi} \cos\theta \, \mathrm{e}^{r_i r'_j \cos\theta/\ell^2} & = I_1\left(\frac{r_i r'_j}{\ell^2}\right) \\ \int_0^{2\pi} \frac{d\theta}{2\pi} \sin\theta \, \mathrm{e}^{r_i r'_j \cos\theta/\ell^2} & = 0 \end{cases}$$

where $I_1(\cdot)$ is a modified Bessel function of the first kind. The second line follows because we are integrating an odd function over an even domain. The first line, on the other hand, results from a definition of modified Bessel functions of the first kind $I_n(z)$ for integer values of n ([138] p. 376), i.e.

$$I_n(z) = \frac{1}{\pi} \int_0^{\pi} \mathrm{e}^{z\cos\theta} \cos(n\theta) d\theta.$$

Hence the final integral reads

$$I_{ij} = C_{ij} I_1\left(\frac{r_i r'_j}{\ell^2}\right) \mathbf{R}_{ij}.$$

3D systems

For notational convenient, one can cast the integral in the following form

$$\begin{aligned} I_{ij} &= \int d\mathcal{R}\, \mathbf{R}\, \mathrm{e}^{-(\mathbf{r}_i - \mathbf{R}\mathbf{r}'_j)^2/2\ell^2} \\ &= \int d\mathcal{R}\, \mathbf{R}\, k_{SE}(\mathbf{r}_i, \mathbf{R}\mathbf{r}'_j). \end{aligned}$$

To start with, the *global invariance* of the base squared exponential pairwise kernels k_{SE}, that is $k_{SE}(\mathbf{r}, \mathbf{r}') = k_{SE}(\mathbf{R}\mathbf{r}, \mathbf{R}\mathbf{r}')$, can be used in order to align $\mathbf{r}_i$ onto the z-axis. Calling the rotation that does so $\mathbf{R}_i^z$ and we have

$$\begin{aligned} I_{ij} &= \int d\mathcal{R}\, \mathbf{R}\, k_{SE}(\mathbf{R}_i^z \mathbf{r}_i, \mathbf{R}_i^z \mathbf{R}\mathbf{r}'_j) \\ &= \int d\mathcal{R}\, \mathbf{R}\, k_{SE}(\tilde{\mathbf{r}}_i, \mathbf{R}_i^z \mathbf{R}\mathbf{r}'_j). \end{aligned}$$

where we defined $\tilde{\mathbf{r}}_i = \mathbf{R}_i^z \mathbf{r}_i$. At this point we find the matrix $\mathbf{R}_j^z$ that brings also $\mathbf{r}_j$ parallel to the z-axis. We then insert it in front of $\mathbf{r}'_j$ in the form of the

identity $\mathbf{R}_j^{z\mathrm{T}}\mathbf{R}_j^z$:

$$\begin{aligned} I_{ij} &= \int d\mathcal{R}\,\mathbf{R}\,k_{SE}(\tilde{\mathbf{r}}_i, \mathbf{R}_i^z\mathbf{R}\mathbf{R}_j^{z\mathrm{T}}\mathbf{R}_j^z\mathbf{r}'_j) \\ &= \int d\mathcal{R}\,\mathbf{R}\,k_{SE}(\tilde{\mathbf{r}}_i, \mathbf{R}_i^z\mathbf{R}\mathbf{R}_j^{z\mathrm{T}}\tilde{\mathbf{r}}'_j) \end{aligned}$$

where we again used the tilde notation to define the vector now aligned to the z-azis. Finally, the change of variables $\tilde{\mathbf{R}} = \mathbf{R}_i^z\mathbf{R}\mathbf{R}_j^{z\mathrm{T}}$ gives

$$\begin{aligned} I_{ij} &= \mathbf{R}_i^{z\mathrm{T}} \int d\tilde{\mathcal{R}}\,\tilde{\mathbf{R}}\,k_{SE}(\tilde{\mathbf{r}}_i, \tilde{\mathbf{R}}\tilde{\mathbf{r}}'_j)\mathbf{R}_j^z \\ &= \mathbf{R}_i^{z\mathrm{T}}\mathbf{R}_{ij}\mathbf{R}_j^z. \end{aligned}$$

The central integral yielding $\mathbf{R}_{ij}$ remains to be performed. Its evaluation is considerably simpler than the original problem since now both vectors $\tilde{\mathbf{r}}_i, \tilde{\mathbf{r}}'_j$ are along the z-axis. Hence, by parametrising all rotations by Euler angles α, β, γ around the z, y, z axes respectively, we find by geometric reasoning that the argument of the exponential has to be invariant upon rotations of angles α and γ around the z-axis. In fact, we have that

$$\mathbf{R}_{ij} = C_{ij} \int \frac{d\alpha d\beta d\gamma \sin\beta}{8\pi^2} \mathbf{R}(\alpha, \beta, \gamma) \mathrm{e}^{r_i r'_j \cos\beta/\ell^2}$$

where we use the normalised Haar measure $d\alpha d\beta d\gamma \sin\beta/8\pi^3$. The rotation matrix to be averaged reads

$$\mathbf{R}(\alpha, \beta, \gamma) = \begin{pmatrix} c_\alpha c_\gamma - c_\beta s_\alpha s_\gamma & -c_\gamma c_\beta s_\alpha - c_\alpha s_\gamma & s_\alpha s_\beta \\ c_\gamma s_\alpha + c_\alpha c_\beta s_\gamma & c_\alpha c_\gamma c_\beta - s_\alpha s_\gamma & -c_\alpha s_\beta \\ s_\gamma s_\beta & c_\gamma s_\beta & c_\beta \end{pmatrix}.$$

All the elements of the above matrix apart from the zz element vanish since there is always either a sine or a cosine integrated over an entire period. By

defining $\gamma_{ij} = r_i r'_j/\ell^2$, the only non trivial integral reads

$$\begin{aligned}\int_0^\pi \frac{d\beta \sin\beta}{2} \cos\beta \, \mathrm{e}^{r_i r'_j \cos\beta/\ell^2} &= \int_0^\pi \frac{d\beta}{2} \frac{\sin(2\beta)}{2} \mathrm{e}^{\gamma_{ij}\cos\beta} \\ &= \left[\frac{\mathrm{e}^{\gamma_{ij}\cos\beta}(1-\gamma_{ij}\cos\beta)}{2\gamma_{ij}^2}\right]_0^\pi \\ &= \frac{\gamma_{ij}\cosh\gamma_{ij} - \sinh\gamma_{ij}}{\gamma_{ij}^2}.\end{aligned}$$

A.8 Proof that 2-body covariant kernels give rise to central forces

2D systems

Exploiting the decomposition of the $O(2)$ given in Eq. (4.14), the kernel $\mathbf{K}^{O(2)}$ can be written in terms of the already calculated $\mathbf{K}^{SO(2)}$ kernel simply as

$$\mathbf{K}_2^{O(2)}(\rho,\rho') = \frac{1}{2}\left[\mathbf{K}^{SO(2)}(\rho,\rho') + \mathbf{K}^{SO(2)}(\rho,\mathcal{F}\rho')\mathbf{F}\right]$$

To show that the above kernel is equivalent to the "radial" form given in Eq. (4.19), it is convenient to first write the rotation matrix $\mathbf{R}_{ij}$ bringing the unit vector $\hat{\mathbf{r}}_i$ onto $\hat{\mathbf{r}}_j$ as

$$\mathbf{R}_{ij} = \hat{\mathbf{r}}_i\hat{\mathbf{r}}_j'^{\mathrm{T}} + \mathbf{R}_\llcorner\hat{\mathbf{r}}_i\hat{\mathbf{r}}_j'^{\mathrm{T}}\mathbf{R}_\llcorner^{\mathrm{T}},$$

where $\mathbf{R}_\llcorner$ is a 90-degree rotation matrix. Substituting the above into the explicit expression for the $\mathbf{K}^{SO(2)}$ kernel (4.17), the covariant kernel $\mathbf{K}^{O(2)}$ 4.18 reads

$$\mathbf{K}_2^{O(2)}(\rho,\rho') = \frac{1}{2}\sum_{ij}\phi(r_i,r'_j)\left[(\hat{\mathbf{r}}_i\hat{\mathbf{r}}_j'^{\mathrm{T}} + \mathbf{R}_\llcorner\hat{\mathbf{r}}_i\hat{\mathbf{r}}_j'^{\mathrm{T}}\mathbf{R}_\llcorner^{\mathrm{T}}) + (\hat{\mathbf{r}}_i\hat{\mathbf{r}}_j'^{\mathrm{T}}\mathbf{F}^{\mathrm{T}} + \mathbf{R}_\llcorner\hat{\mathbf{r}}_i\hat{\mathbf{r}}_j'^{\mathrm{T}}\mathbf{F}^{\mathrm{T}}\mathbf{R}_\llcorner^{\mathrm{T}})\mathbf{F}\right]$$

Finally, by grouping together the radial terms and using the fact that $\mathbf{R}_{\llcorner}^{\mathrm{T}} + \mathbf{R}_{\llcorner} = 0$ one obtains the desired result

$$\begin{aligned}\mathbf{K}_2^{O(2)}(\rho,\rho') &= \frac{1}{2}\sum_{ij}\phi(r_i,r'_j)\left[\hat{\mathbf{r}}_i\hat{\mathbf{r}}_j'^{\mathrm{T}}(1+\mathbf{F}\mathbf{F}^{\mathrm{T}}) + \mathbf{R}_{\llcorner}\hat{\mathbf{r}}_i\hat{\mathbf{r}}_j'^{\mathrm{T}}(\mathbf{R}_{\llcorner}^{\mathrm{T}} + \mathbf{F}^{\mathrm{T}}\mathbf{R}_{\llcorner}^{\mathrm{T}}\mathbf{F})\right] \\ &= \frac{1}{2}\sum_{ij}\phi(r_i,r'_j)\left[2\,\hat{\mathbf{r}}_i\hat{\mathbf{r}}_j'^{\mathrm{T}} + \mathbf{R}_{\llcorner}\hat{\mathbf{r}}_i\hat{\mathbf{r}}_j'^{\mathrm{T}}(\mathbf{R}_{\llcorner}^{\mathrm{T}} + \mathbf{R}_{\llcorner})\right] \\ &= \sum_{ij}\phi(r_i,r'_j)\,\hat{\mathbf{r}}_i\hat{\mathbf{r}}_j'^{\mathrm{T}}.\end{aligned}$$

3D systems

To show that the $\mathbf{K}^{SO(3)}$ covariant kernel in Eq. (4.20) is equivalent to the radial form given in Eq. (4.21), one first needs to be able to see that the outer product of the $\hat{\mathbf{z}}$ unit vector is

$$\hat{\mathbf{z}}\hat{\mathbf{z}}^{\mathrm{T}} = \begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 1\end{pmatrix}$$

so that Eq. (4.20) can be rewritten as

$$\mathbf{K}_2^{SO(3)}(\rho,\rho') = \sum_{ij}\phi(r_i,r'_j)\mathbf{R}_i^{z\mathrm{T}}\hat{\mathbf{z}}\hat{\mathbf{z}}^{\mathrm{T}}\mathbf{R}_j^{z}.$$

At this point one can exploit the definitions $\mathbf{R}_i^z\hat{\mathbf{r}}_i = \hat{\mathbf{z}}$ and $\mathbf{R}_j^z\hat{\mathbf{r}}_j = \hat{\mathbf{z}}$ to find that $\mathbf{R}_i^{z\mathrm{T}}\hat{\mathbf{z}} = \hat{\mathbf{r}}_i$ and $(\mathbf{R}_j^z\hat{\mathbf{z}})^{\mathrm{T}} = \hat{\mathbf{r}}_j^{\mathrm{T}}$, obtaining this way the final result

$$\begin{aligned}\mathbf{K}_2^{SO(3)}(\rho,\rho') &= \sum_{ij}\phi(r_i,r'_j)\hat{\mathbf{r}}_i\hat{\mathbf{r}}_j^{\mathrm{T}} \\ &= \mathbf{K}_2^{O(3)}(\rho,\rho'),\end{aligned}$$

where the last equality can be easily checked.

A.9 Covariant integration of 3-body kernels

The covariant integral for a 3-body kernel in two and three spatial dimensions ($d \in \{2, 3\}$) is

$$\begin{aligned} \mathbf{K}_3^{SO(d)} &= \sum_{ij}\sum_{lm} \int dR\, \mathbf{R} \mathrm{e}^{-(\mathbf{r}_i - \mathbf{R}\mathbf{r}'_j)^2/2\ell^2} \mathrm{e}^{-(\mathbf{r}_l - \mathbf{R}\mathbf{r}'_m)^2/2\ell^2} \\ &= \sum_{ij}\sum_{lm} I_{ijlm}. \end{aligned}$$

Separating the radial part from the angular one, I_{ijlm} reads:

$$\begin{aligned} I_{ijlm} &= \mathrm{e}^{-(r_i^2 + r_j'^2 + r_l^2 + r_m'^2)/2\ell^2} \int dR\, \mathbf{R}\, \mathrm{e}^{\mathbf{r}_i^{\mathrm{T}}\mathbf{R}\mathbf{r}'_j/\ell^2} \mathrm{e}^{\mathbf{r}_l^{\mathrm{T}}\mathbf{R}\mathbf{r}'_m/\ell^2} \\ &= \tilde{C}_{ijlm} \int dR\, \mathbf{R}\, \mathrm{e}^{\frac{1}{\ell^2}[\mathbf{r}_i^{\mathrm{T}}\mathbf{R}\mathbf{r}'_j + \mathbf{r}_l^{\mathrm{T}}\mathbf{R}\mathbf{r}'_m]} \end{aligned}$$

2D systems

In two dimensions the integral can be written as

$$\begin{aligned} I_{ijlm} &= \tilde{C}_{ijlm} \int \frac{d\theta}{2\pi}\, \mathbf{R}(\theta)\, \mathrm{e}^{\frac{1}{\ell^2}[r_i r'_j \cos(\theta_{ij}+\theta) + r_l r'_m \cos(\theta_{lm}+\theta)]} \\ &= \tilde{C}_{ijlm} \int \frac{d\theta}{2\pi}\, \mathbf{R}(\theta)\, \mathrm{e}^{[A_{ij}\cos(\theta_{ij}+\theta) + A_{lm}\cos(\theta_{lm}+\theta)]}. \end{aligned}$$

It is now possible to simplify the argument of the exponential by recurring to standard trigonometric identities. The two terms appearing in the argument in the exponential can be written as

$$\begin{aligned} A_{ij}\cos(\theta_{ij}+\theta) &= A_{ij}\cos\theta_{ij}\cos\theta - A_{ij}\sin\theta_{ij}\sin\theta \\ A_{lm}\cos(\theta_{lm}+\theta) &= A_{lm}\cos\theta_{lm}\cos\theta - A_{lm}\sin\theta_{lm}\sin\theta \end{aligned}$$

and their sum can be hence be brought to

$$\begin{aligned} &(A_{ij}\cos\theta_{ij} + A_{lm}\cos\theta_{lm})\cos\theta \\ &\qquad - (A_{ij}\sin\theta_{ij} + A_{lm}\sin\theta_{lm})\sin\theta = C_{ijlm}\cos\theta - S_{ijlm}\sin\theta. \end{aligned}$$

The two amplitude parameters in the last expression can be finally cast into an amplitude and a phase as follows

$$\begin{aligned} C_{ijlm}\cos\theta - S_{ijlm}\sin\theta &= \sqrt{C_{ijlm}^2 + S_{ijlm}^2}\cos(\theta + \tan^{-1}(\frac{S_{ijlm}}{C_{ijlm}})) \\ &= A_{ijlm}\cos(\theta + \theta_{ijlm}). \end{aligned}$$

In this form, the integral can be evaluated as

$$\begin{aligned} I_{ijlm} &= \tilde{C}_{ijlm}\int \frac{d\theta}{2\pi}\,\mathbf{R}(\theta)\,\mathrm{e}^{A_{ijlm}\cos(\theta+\theta_{ijlm})} \\ &= \tilde{C}_{ijlm}\mathbf{R}(\theta_{ijlm})\int \frac{d\theta}{2\pi}\,\mathbf{R}(\theta)\,\mathrm{e}^{A_{ijlm}\cos(\theta)} \\ &= \tilde{C}_{ijlm}\mathbf{R}(\theta_{ijlm})I_1(A_{ijlm}), \end{aligned}$$

where from the first to the second line the change of variables $\theta + \theta_{ijlm} \to \theta$ was performed and I_1 is a modified Bessel function of the first kind.

Hence, the final result for the second order covariant kernel in two dimension is

$$\begin{aligned} \mathbf{K}_3^{SO(2)} &= \sum_{ijlm}\tilde{C}_{ijlm}I_1(A_{ijlm})\mathbf{R}(\theta_{ijlm}) \\ \tilde{C}_{ijlm} &= \mathrm{e}^{-(r_i^2+r_j'^2+r_l^2+r_m'^2)/2\ell^2} \\ \theta_{ijlm} &= \tan^{-1}\left(\frac{r_i r_j'\sin\theta_{ij} + r_l r_m'\sin\theta_{lm}}{r_i r_j'\cos\theta_{ij} + r_l r_m'\cos\theta_{lm}}\right) \\ A_{ijlm} &= \sqrt{(r_i r_j'\cos\theta_{ij} + r_l r_m'\cos\theta_{lm})^2 + (r_i r_j'\sin\theta_{ij} + r_l r_m'\sin\theta_{lm})^2}/\ell^2 \end{aligned}$$

Notice that the argument of the Bessel function can be rewritten in terms of the angles "within" each configuration (i.e. $\theta_{il} \wedge i, l \in \rho$ and $\theta_{jm} \wedge j, m \in \rho'$) as

$$A_{ijlm} = \sqrt{(r_i r_j)^2 + (r_l r_m)^2 + 2r_i r_j r_l r_m\cos(\theta_{il} - \theta_{jm})}/\ell^2.$$

3D systems

As in the 2-body case, for notational convenience the relevant integral is rewritten as

$$\begin{aligned} I_{ijlm} &= \int dR\, \mathbf{R} \mathrm{e}^{-(\mathbf{r}_i - \mathbf{R}\mathbf{r}'_j)^2/2\ell^2} \mathrm{e}^{-(\mathbf{r}_l - \mathbf{R}\mathbf{r}'_m)^2/2\ell^2} \\ &= \int dR\, \mathbf{R}\, k_{SE}(\mathbf{r}_i, \mathbf{R}\mathbf{r}'_j) k_{SE}(\mathbf{r}_l, \mathbf{R}\mathbf{r}'_m). \end{aligned}$$

The global invariance of the pairwise squared exponential kernels k_{SE}, that is $k_{SE}(\mathbf{r}, \mathbf{r}') = k_{SE}(\mathbf{R}\mathbf{r}, \mathbf{R}\mathbf{r}')$, can be now exploited to align the first couple of atoms $(\mathbf{r}_i, \mathbf{r}_l)$ in such a way that $\mathbf{r}_i$ is parallel to the z-axis and $\mathbf{r}_l$ lies in the xz plane. ZWe call the rotation that does so $\mathbf{R}_{il}$ and we have

$$\begin{aligned} I_{ijlm} &= \int dR\, \mathbf{R}\, k_{SE}(\mathbf{R}_{il}\mathbf{r}_i, \mathbf{R}_{il}\mathbf{R}\mathbf{r}'_j) k_{SE}(\mathbf{R}_{il}\mathbf{r}_l, \mathbf{R}_{il}\mathbf{R}\mathbf{r}'_m) \\ &= \int dR\, \mathbf{R}\, k_{SE}(\tilde{\mathbf{r}}_i, \mathbf{R}_{il}\mathbf{R}\mathbf{r}'_j) k_{SE}(\tilde{\mathbf{r}}_l, \mathbf{R}_{il}\mathbf{R}\mathbf{r}'_m). \end{aligned}$$

At this point we find the matrix $\mathbf{R}_{jm}$ that bring $\mathbf{r}_j$ parallel to the z-axis and $\mathbf{r}_m$ onto the xz plane. We write

$$\begin{aligned} I_{ijlm} &= \int dR\, \mathbf{R}\, k_{SE}(\tilde{\mathbf{r}}_i, \mathbf{R}_{il}\mathbf{R}\mathbf{R}^{\mathrm{T}}_{jm}\mathbf{R}_{jm}\mathbf{r}'_j) k_{SE}(\tilde{\mathbf{r}}_l, \mathbf{R}_{il}\mathbf{R}\mathbf{R}^{\mathrm{T}}_{jm}\mathbf{R}_{jm}\mathbf{r}'_m) \\ &= \int dR\, \mathbf{R}\, k_{SE}(\tilde{\mathbf{r}}_i, \mathbf{R}_{il}\mathbf{R}\mathbf{R}^{\mathrm{T}}_{jm}\tilde{\mathbf{r}}'_j) k_{SE}(\tilde{\mathbf{r}}_l, \mathbf{R}_{il}\mathbf{R}\mathbf{R}^{\mathrm{T}}_{jm}\tilde{\mathbf{r}}'_m). \end{aligned}$$

Finally we perform the change of variables $\tilde{\mathbf{R}} = \mathbf{R}_{il}\mathbf{R}\mathbf{R}^{\mathrm{T}}_{jm}$ to obtain

$$\begin{aligned} I_{ijlm} &= \mathbf{R}^{\mathrm{T}}_{il} \int dR\, \tilde{\mathbf{R}}\, k_{SE}(\tilde{\mathbf{r}}_i, \tilde{\mathbf{R}}\tilde{\mathbf{r}}'_j) k_{SE}(\tilde{\mathbf{r}}_l, \tilde{\mathbf{R}}\tilde{\mathbf{r}}'_m) \mathbf{R}_{jm} \\ &= \mathbf{R}^{\mathrm{T}}_{il} \mathbf{R}_{ijlm} \mathbf{R}_{jm}. \end{aligned}$$

The central integral yielding $\mathbf{R}_{ijlm}$ remains to be performed. Exact analytical solution for $\mathbf{R}_{ijlm}$ is difficult to obtain. However, we can find a very good analytical approximation to it.

We start by noticing that once the vectors are oriented as explained above, the maximum of the integrand will certainly lie within the xz plane. This means that, by choosing the Tait-Bryan angles α, β, γ to be given by the sequence of rotations around $y - z - x$ respectively the maximum will be at

$\alpha_0 = \theta_{ijlm}, \beta_0 = 0, \gamma_0 = 0$. Moreover, given the above mentioned alignments, we have that $\theta_{ij} = 0$ and the alignment angle α_0 will be given by

$$\alpha_0 = \tan^{-1}\left(\frac{r_l r'_m \sin\theta_{lm}}{r_i r'_j + r_l r'_m \cos\theta_{lm}}\right).$$

As we found a maximum, we can Taylor expand the angular part in the exponential around that point to second order. For convenience let us define $\boldsymbol{\theta} \equiv (\alpha, \beta, \gamma)^{\mathrm{T}}$ the angular part to be expanded reads

$$f(\boldsymbol{\theta}) \equiv \frac{1}{2\sigma^2}[\tilde{\mathbf{r}}_i^{\mathrm{T}}\tilde{\mathbf{R}}(\alpha,\beta,\gamma)\tilde{\mathbf{r}}'_j + \tilde{\mathbf{r}}_l^{\mathrm{T}}\tilde{\mathbf{R}}(\alpha,\beta,\gamma)\tilde{\mathbf{r}}'_m]$$

so that

$$f(\boldsymbol{\theta}) \approx f(\boldsymbol{\theta}_0) - \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^{\mathrm{T}}\left[-\frac{\partial^2 f(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\mathrm{T}}}\right]_{\boldsymbol{\theta}_0}(\boldsymbol{\theta}-\boldsymbol{\theta}_0).$$

The calculation of the (negative) Hessian $\mathbf{H} = -\left[\frac{\partial^2 f}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\mathrm{T}}}\right]_{\boldsymbol{\theta}_0}$ proceeds as follows. First of all let us write down explicitly the analytic dependence of $f(\boldsymbol{\theta})$ on the Tait-Bryan angles. A generic rotation matrix around the y–z–x angles is written down as

$$\mathbf{R}^{yzx}(\alpha,\beta,\gamma) = \begin{pmatrix} c_\alpha c_\beta & -s_\beta & c_\beta s_\alpha \\ c_\alpha c_\gamma s_\beta + s_\alpha s_\gamma & c_\beta c_\gamma & c_\gamma s_\alpha s_\beta - c_\alpha s_\gamma \\ c_\alpha s_\beta s_\gamma - c_\gamma s_\alpha & c_\beta s_\gamma & c_\alpha c_\gamma + s_\alpha s_\beta s_\gamma \end{pmatrix},$$

where c_α (s_α) is a shorthand notation for the cosine (sine) of the angle α. One can then write down the sum of quadratic forms

$$f(\boldsymbol{\theta}) \;=\; \frac{1}{\ell^2}\left[\begin{pmatrix}0 & 0 & r_i^z\end{pmatrix}\mathbf{R}^{yzx}\begin{pmatrix}0\\0\\r'^z_j\end{pmatrix} + \begin{pmatrix}r_l^x & 0 & r_l^z\end{pmatrix}\mathbf{R}^{yxz}\begin{pmatrix}r_l^x\\0\\r'^z_j\end{pmatrix}\right],$$

which reads

$$f(\boldsymbol{\theta}) = \frac{1}{\ell^2}[c_\alpha(r_l^z r'^x_m s_\beta s_\gamma + r_l^x r'^x_m c_\beta + r_i^z r'^z_j c_\gamma + r_l^z r'^z_m c_\gamma) \\ + s_\alpha(s_\beta s_\gamma(r_i^z r'^z_j + r_l^z r'^z_m) + r_l^x r'^z_m c_\beta - r_l^z r'^x_m c_\gamma)].$$

It is then a tedious exercise to calculate, one by one, all the double derivatives

$\frac{\partial^2 f}{\partial\theta_i\partial\theta_j}$ and evaluate them at the point $\boldsymbol{\theta}_0 = \begin{pmatrix}\alpha_0 & 0 & 0\end{pmatrix}^{\mathrm{T}}$. The result reads

$$\mathbf{H} = -\begin{pmatrix} H_{\alpha\alpha} & 0 & 0 \\ 0 & H_{\beta\beta} & H_{\beta\gamma} \\ 0 & H_{\beta\gamma} & H_{\gamma\gamma}\end{pmatrix}$$

$$\begin{aligned} H_{\alpha\alpha} &= [(r_l^z r_m'^x - r_l^x r_m'^z)s_{\alpha_0} - (r_i^z r_j'^z + r_l^x r_m'^x + r_l^z r_m'^z)c_{\alpha_0}]/\ell^2 \\ H_{\beta\beta} &= [-r_l^x(r_m'^x c_{\alpha_0} + r_m'^z s_{\alpha_0})]/\ell^2 \\ H_{\gamma\gamma} &= [r_l^z r_m'^x s_{\alpha_0} - (r_i^z r_j'^z + r_l^z r_m'^z)c_{\alpha_0}]/\ell^2 \\ H_{\beta\gamma} &= [r_l^z r_m'^x c_{\alpha_0} + (r_i^z r_j'^z + r_l^z r_m'^z)s_{\alpha_0}]/\ell^2. \end{aligned}$$

After the above manipulations, the original integral over all rotations has been transformed into an expected value of a rotation matrix over a multivariate normal distribution:

$$\begin{aligned} \mathbf{R}_{ijlm} &= \tilde{C}_{ijlm}\mathrm{e}^{f_{ijlm}} \int d\boldsymbol{\theta}\, \mathbf{R}(\boldsymbol{\theta})\, \mathrm{e}^{-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^{\mathrm{T}}\mathbf{H}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)} \\ &= \tilde{C}_{ijlm}\mathrm{e}^{f_{ijlm}} Z_{ijlm} \left\langle \mathbf{R}(\boldsymbol{\theta})\right\rangle_{\mathcal{N}(\boldsymbol{\theta}_0, \mathbf{H}^{-1})} \end{aligned}$$

where Z is the normalisation of the probability distribution, given by

$$Z_{ijlm} = \sqrt{\frac{|\ \mathbf{H}_{ijlm}\ |}{(2\pi)^3}}$$

while the quadratic form f_{ijlm} is written explicitly as

$$f_{ijlm} = \frac{1}{\ell^2}[c_{\alpha_0}(r_l^x r_m'^x + r_i^z r_j'^z + r_l^z r_m'^z) + s_{\alpha_0}(r_l^x r_m'^z - r_l^z r_m'^x)].$$

The expected value of the rotation matrix is taken element-wise. The structure of all the integrals is similar, hence, all 9 of them are evaluated in the same way. For instance, the xx element of $\mathbf{R}^{yzx}(\alpha,\beta,\gamma)$ reads $\cos\alpha\cos\beta$. First of all, using prostapheresis formulas we are able to rewrite this term as $(\cos(\alpha-\beta) - \cos(\alpha+\beta))/2$. At this point, the random variable of interest is the sum or the difference of correlated gaussian random variables (α and β). We know that $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\theta}_0, \mathbf{H}^{-1})$ hence the variable $x = \alpha \pm \beta$ is also normally distributed $x \sim \mathcal{N}(\mu_x, \sigma_x^2)$. We are hence left with the calculations of $\langle\cos(x)\rangle$. This is easily obtained using Euler's formula and the characteristic function of

the normal distribution as follows:

$$\begin{aligned}\langle \mathrm{e}^{ix}\rangle &= \langle\cos(x)\rangle + i\langle\sin x\rangle \\ &= \mathrm{e}^{i\mu_x-\frac{1}{2}\sigma_x^2} \\ &= (\cos\mu_x + i\sin\mu_x)\mathrm{e}^{-\frac{1}{2}\sigma_x^2}\end{aligned}$$

$$\Downarrow$$

$$\begin{cases}\langle\cos x\rangle &= \mathrm{e}^{-\frac{1}{2}\sigma_x^2}\cos\mu_x \\ \langle\sin x\rangle &= \mathrm{e}^{-\frac{1}{2}\sigma_x}\sin\mu_x\end{cases}$$

where to calculate the expected value of the exponential $\langle \mathrm{e}^{ix}\rangle$ we used the characteristic function of the normal distribution $\langle \mathrm{e}^{itx}\rangle = \mathrm{e}^{it\mu_x-\frac{1}{2}t^2\sigma_x^2}$ evaluated at $t = 1$. In the following, the result of the nine integrals (all evaluated as explained above) is reported. These compose the expected value of the whole rotation matrix

$$\begin{aligned}\langle\mathbf{R}(\boldsymbol{\theta})\rangle_{\mathcal{N}(\boldsymbol{\theta}_0,\mathbf{H}^{-1})} &\equiv \hat{\mathbf{R}}_{ijlm} \\ &= \begin{pmatrix}R_{xx} & 0 & R_{xz}\\ 0 & R_{yy} & 0\\ R_{zx} & 0 & R_{zz}\end{pmatrix}\end{aligned}$$

$$\begin{aligned}R_{xx} &= \mathrm{e}^{-\frac{1}{2}(H_{\alpha\alpha}+H_{\beta\beta})}\cos\alpha_0 \\ R_{yy} &= \mathrm{e}^{-\frac{1}{2}(H_{\beta\beta}+H_{\gamma\gamma})}\cosh H_{\beta\gamma} \\ R_{zz} &= \mathrm{e}^{-\frac{1}{2}(H_{\alpha\alpha}+H_{\beta\beta}+H_{\gamma\gamma})}(\sinh H_{\beta\gamma}\sin\alpha_0 + \mathrm{e}^{\frac{1}{2}H_{\beta\beta}}\cos\alpha_0) \\ R_{xz} &= \mathrm{e}^{-\frac{1}{2}(H_{\alpha\alpha}+H_{\beta\beta})}\sin\alpha_0 \\ R_{zx} &= \mathrm{e}^{-\frac{1}{2}(H_{\alpha\alpha}+H_{\beta\beta})}(\sinh H_{\beta\gamma}\cos\alpha_0 - \mathrm{e}^{\frac{1}{2}H_{\beta\beta}}\sin\alpha_0)\end{aligned}$$

Hence, the final integral over all three dimensional rotations reads:

$$I_{ijlm} \approx \tilde{C}_{ijlm}\mathrm{e}^{f_{ijlm}}Z_{ijlm}\mathbf{R}_{ij}^{\mathrm{T}}\hat{\mathbf{R}}_{ijlm}\mathbf{R}_{lm}.$$

Although approximate, the result just derived has the nice property of becoming more accurate as $\ell \to 0$, as this will make the relevant integral more peaked around the exactly computable maximum. Since, typically ℓ is chosen to be relatively small compared to the scale of the (as this helps in resolving the

atomic environments), one can expect the approximation to work reasonably well in practice.

A.10 A second one dimensional toy model

The fictitious n-body interaction model on which model selection ideas were tested was set up as a hierarchy of two body interactions defined via the following negative Gaussian

$$\epsilon^g(d) = -\mathrm{e}^{-\frac{k(d-a)^2}{2}},$$

where a and k can be thought to model a lattice parameter and a spring constant.

This pairwise interaction, depending only on the distance d between two particles, was then used to generate n-body local energies as

$$\epsilon_n(\rho) = \sum_{i_1 \neq \cdots \neq i_{n-1}} \epsilon^g(x_{i_1})\epsilon^g(x_{i_2} - x_{i_1}) \ldots \epsilon^g(x_{i_{n-2}} - x_{i_{n-1}})$$

where $x_{i_1}, \ldots, x_{i_n - 1}$ are the positions, relative to the central atom, of $n-1$ surrounding particles.

A.11 Mapping the predictive variance

To gain insights on the problems of mapping the predictive variance, it is instructive to look at the simples example of a 2-body kernel.

In such a case, the equation for the predicted variance, obtained substituting the kernel definition of Eq. (3.2) in the third expression of Eq. (2.7) is

$$\hat{\sigma}^2(\rho) = \sum_{i,k \in \rho} \left(\mathrm{e}^{-(r_i - r_k)^2/2\ell^2} - \sum_{t,u=1}^{N} \sum_{\substack{j \in \rho_t \\ l \in \rho_u}} \mathrm{e}^{-(r_i - r_j)^2/2\ell^2} (\boldsymbol{C}^{-1})_{tu} \mathrm{e}^{-(r_k - r_l)^2/2\ell^2} \right).$$

The above be rewritten by defining the part in parenthesis to be the function

$\tilde{\sigma}^2(r_i, r_k)$, obtaining:

$$\hat{\sigma}^2(\rho) = \sum_{i,k \in \rho} \tilde{\sigma}^2(r_i, r_k).$$

Similarly to what was done for the predicted mean, the values of the function $\tilde{\sigma}^2(r_i, r_k)$ can be stored and locally interpolated.

However, differently from the predicted mean, the predicted variance is a function of two variables ($(3n-6)^2$ variables for a general n-body kernel with $n = 2$). In general, the dimensionality of the n-body GP mapping hence increases from $3n - 6$ (for the mean prediction) to $(3n-6)^2$. This makes mapping the variance a very cumbersome operation already for $n = 3$.

In such a case, the mapping of the GP predicted error for can be made computationally affordable by approximating the error contribution from each n-plet to be independent. In the practical example provided above this would correspond to assume a predicted error of the form

$$\hat{\sigma}^2(\rho) = \sum_{i \in \rho} \tilde{\sigma}^2(r_i, r_i),$$

where all non-diagonal contributions with $r_i \neq r_k$ have been neglected and only the one dimensional function $\tilde{\sigma}^2(r_i, r_i)$ needs to be mapped. This alternative measure of predictive uncertainty is unlikely to be close to the original one (since the cross terms in the covariance are unlikely going to be zero). However, it represents a valid and meaningful error estimate and its accuracy should be tested directly on reference systems.

A.12 Quadratic scaling of the complete kernel

Let us assume we have a database configuration $\mathbf{x}_d$ and a root configuration $\mathbf{x}_r$. Omitting unessential factors, the complete kernel among those two is written down as

$$k_c(\mathbf{x}_r, \mathbf{x}_d) \sim \sum_{i,j=1}^{L} (1 + \sum_{\Delta} \delta_{x^r_{i+\Delta}, x^d_{j+\Delta}})^{L-1}.$$

Then let us assume that a new configuration $\mathbf{x}_n$ is a neighbour of the root $\mathbf{x}_r$ (i.e. $\mathbf{x}_n \in \partial \mathbf{x}_r$) such that there will be only two indices of $\mathbf{x}_n$ that change with respect to those of $\mathbf{x}_r$. Let us define by $\mathcal{C}$ the set of the two changed indices and by $\bar{\mathcal{C}}$ the set of $L-2$ unchanged ones.

Finally, let us define the $L \times L$ array containing the result of the delta function evaluations along the displacements between the root and the database configuration

$$A_{ij}^{rd} = \sum_{\Delta} \delta_{x^r_{i+\Delta}, x^d_{j+\Delta}}.$$

Then, by noticing that the delta function operator will yield identical results on most of the new configuration

$$\delta_{x^x_{i+\Delta}, x^d_{j+\Delta}} = \delta_{x^r_{i+\Delta}, x^d_{j+\Delta}} \quad \forall \Delta \mid i + \Delta \in \bar{\mathcal{C}},$$

it is simple to see that the array A_{ij}^{nd} for the new configuration can be calculated in order $\mathcal{O}(1)$ as

$$\begin{aligned} A_{ij}^{nd} &= \sum_{\Delta} \delta_{x^n_{i+\Delta}, x^d_{j+\Delta}} \\ &= \sum_{\Delta \in \bar{\mathcal{C}}_i} \delta_{x^x_{i+\Delta}, x^d_{j+\Delta}} + \sum_{\Delta \in \mathcal{C}_i} \delta_{x^x_{i+\Delta}, x^d_{j+\Delta}} \\ &= A_{ij}^{rd} - \sum_{\Delta \in \bar{\mathcal{C}}_i} \delta_{x^r_{i+\Delta}, x^d_{j+\Delta}} + \sum_{\Delta \in \mathcal{C}_i} \delta_{x^x_{i+\Delta}, x^d_{j+\Delta}} \end{aligned}$$

where $\mathcal{C}_i$ (and $\bar{\mathcal{C}}_i$) are the set of indices for which $i + \Delta \in \mathcal{C} (\in \bar{\mathcal{C}})$ respectively.

This brings the scaling of the kernel evaluations $k_c(\mathbf{x}_r, \mathbf{x}_d)$ to L^2 at the cost of storing and updating the array A_{ij}^{rd} for a root configuration r and each one of the N database entries d.

Bibliography

[1] Harding, L. & Barden, L. Deep Blue win a giant step for computerkind. *The Guardian* (1997). URL `https://www.theguardian.com/theguardian/2011/may/12/deep-blue-beats-kasparov-1997`.

[2] Borowiec, S. Google's AlphaGo AI defeats human in first game of Go contest. *The Guardian* (2016). URL `https://www.theguardian.com/technology/2016/mar/09/google-deepmind-alphago-ai-defeats-human-lee-sedol-first-game-go-contest`.

[3] Russell, S. & Norvig, P. *Artificial Intelligence: A Modern Approach* (Pearson, 1995).

[4] Hsu, F. H. IBM's deep blue chess grandmaster chips. *IEEE Micro* **19**, 70–81 (1999).

[5] Hsu, F. H. *Behind Deep Blue: Building the computer that defeated the world chess champion* (Princeton University Press, 2004).

[6] Silver, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).

[7] Silver, D. *et al.* A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* **362**, 1140–1144 (2018).

[8] Dirac, P. A. M. Quantum mechanics of many-electron systems. *Proceedings of the Royal Society A* **123**, 714–733 (1929).

[9] Kohn, W. & Sham, L. J. Self-consistent equations including exchange and correlation effects. *Physical review* **140**, A1133–A1138 (1965).

[10] Hohenberg, P. & Kohn, W. Inhomogeneous electron gas. *Physical review* (1964).

[11] Kantorovich, L. *Quantum Theory of the Solid State: An Introduction* (Springer, 2004).

[12] Marques, M. A. L., Oliveira, M. J. T. & Burnus, T. Libxc: A library of exchange and correlation functionals for density functional theory. *Computer Physics Communications* **183**, 2272–2281 (2012).

[13] Kryder, M. H. & Kim, C. S. After hard drives—what comes next? *IEEE Transactions on Magnetics* **45**, 3406–3413.

[14] Grochowski, E. & Halem, R. D. Technological impact of magnetic hard disk drives on storage systems. *IBM Systems Journal* **42**, 338–346 (2003).

[15] Behler, J. & Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Physical Review Letters* **98**, 146401–4 (2007).

[16] Bartók, A. P., Payne, M. C., Kondor, R. & Csányi, G. Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons. *Physical Review Letters* **104**, 136403–4 (2010).

[17] Snyder, J. C., Rupp, M., Hansen, K., Müller, K. R. & Burke, K. Finding Density Functionals with Machine Learning **108**, 253002–5 (2012).

[18] Ghiringhelli, L. M., Vybiral, J., Levchenko, S. V., Draxl, C. & Scheffler, M. Big Data of Materials Science: Critical Role of the Descriptor. *Physical Review Letters* **114**, 105503–5 (2015).

[19] Geiger, P. & Dellago, C. Neural networks for local structure detection in polymorphic systems. *The Journal of Chemical Physics* **139**, 164105–15 (2013).

[20] De, S., Bartók, A. P., Csányi, G. & Ceriotti, M. Comparing molecules and solids across structural and alchemical space. *Physical Chemistry Chemical Physics* **18**, 13754–13769 (2016).

[21] Stecher, T., Bernstein, N. & Csányi, G. Free Energy Surface Reconstruction from Umbrella Samples Using Gaussian Process Regression. *Journal of Chemical Theory and Computation* **10**, 4079–4097 (2014).

[22] Arsenault, L. F., Lopez-Bezanilla, A., von Lilienfeld, O. A. & Millis, A. J. Machine learning for many-body physics: The case of the Anderson impurity model. *Physical Review B* **90**, 155136–16 (2014).

[23] Stillinger, F. H. & Weber, T. A. Computer simulation of local order in condensed phases of silicon. *Physical review* **B31**, 5262–5271 (1985).

[24] Tersoff, J. New empirical approach for the structure and energy of covalent systems. *Physical Review B* **37**, 6991–7000 (1988).

[25] Brenner, D. W. The art and science of an analytic potential. *Physica Status Solidi B* **217**, 23–40 (2000).

[26] Mishin, Y. Atomistic modeling of the γ and γ'-phases of the Ni–Al system. *Acta Materialia* **52**, 1451–1467 (2004).

[27] van Duin, A. C. T., Dasgupta, S., Lorant, F. & Goddard, W. A. ReaxFF: A Reactive Force Field for Hydrocarbons. *The Journal of Physical Chemistry A* **105**, 9396–9409 (2001).

[28] Cisneros, G. A. *et al.* Modeling Molecular Interactions in Water: From Pairwise to Many-Body Potential Energy Functions. *Chemical Reviews* **116**, 7501–7528 (2016).

[29] Reddy, S. K. *et al.* On the accuracy of the MB-pol many-body potential for water: Interaction energies, vibrational frequencies, and classical thermodynamic and dynamical properties from clusters to liquid water and ice. *The Journal of Chemical Physics* **145**, 194504–14 (2016).

[30] Li, Z., Kermode, J. R. & De Vita, A. Molecular Dynamics with On-the-Fly Machine Learning of Quantum-Mechanical Forces. *Physical Review Letters* **114**, 096405–5 (2015).

[31] Botu, V. & Ramprasad, R. Learning scheme to predict atomic forces and accelerate materials simulations. *Physical Review B* **92**, 094306–5 (2015).

[32] Bartók, A. P., Kondor, R. & Csányi, G. On representing chemical environments. *Physical Review B* **87**, 184115–16 (2013).

[33] Thompson, A. P., Swiler, L. P., Trott, C. R., Foiles, S. M. & Tucker, G. J. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. *Journal of Computational Physics* **285**, 316–330 (2015).

[34] Boes, J. R., Groenenboom, M. C., Keith, J. A. & Kitchin, J. R. Neural network and ReaxFF comparison for Au properties. *International Journal of Quantum Chemistry* **116**, 979–987 (2016).

[35] Sorella, S. Wave function optimization in the variational Monte Carlo method. *Physical Review B* **71**, 241103–4 (2005).

[36] Sorella, S. Generalized Lanczos algorithm for variational quantum Monte Carlo. *Physical Review B* **64**, 973–16 (2001).

[37] Schwarz, L. R., Alavi, A. & Booth, G. H. Projector Quantum Monte Carlo Method for Nonlinear Wave Functions. *Physical Review Letters* **118**, 372–6 (2017).

[38] Mezzacapo, F., Schuch, N., Boninsegni, M. & Cirac, J. I. Ground-state properties of quantum many-body systems: entangled-plaquette states and variational Monte Carlo. *New Journal of Physics* **11**, 083026–10 (2009).

[39] Drummond, N. D., Towler, M. D. & Needs, R. J. Jastrow correlation factor for atoms, molecules, and solids. *Physical Review B* **70**, 12–11 (2004).

[40] Carleo, G. & Troyer, M. Solving the Quantum Many-Body Problem with Artificial Neural Networks. *Science* **365**, 602–606 (2017).

[41] Cai, Z. & Liu, J. Approximating quantum many-body wave functions using artificial neural networks. *Physical Review B* **97**, 035116 (2018).

[42] Ferré, G., Maillet, J. B. & Stoltz, G. Permutation-invariant distance between atomic configurations. *The Journal of Chemical Physics* **143**, 104114–13 (2015).

[43] Glielmo, A., Zeni, C. & De Vita, A. Efficient nonparametric n-body force fields from machine learning. *Physical Review B* **97**, 1–12 (2018).

[44] Kohn, W. Density functional and density matrix method scaling linearly with the number of atoms. *Physical Review Letters* **76**, 3168–3171 (1996).

[45] Prodan, E. & Kohn, W. Nearsightedness of electronic matter. *Proceedings of the National Academy of Sciences* **102**, 11635–11638 (2005).

[46] van Kampen, N. G. *Stochastic Processes in Physics and Chemistry* (Elsevier, 1981).

[47] Rasmussen, C. E. & Williams, C. K. I. *Gaussian Processes for Machine Learning* (The MIT Press, 2006).

[48] Bishop, C. M. *Pattern Recognition and Machine Learning* (Springer, 2006).

[49] Alvarez, M. A., Rosasco, L. & Lawrence, N. D. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning* **4**, 195 (2012).

[50] Micchelli, C. A. & Pontil, M. On learning vector-valued functions. *Neural computation* **17**, 177–204 (2005).

[51] Micchelli, C. A. & Pontil, M. Kernels for multi-task learning. In *Advances in Neural Information Processing Systems 17*, 921–928 (2005).

[52] Feynman, R. P. Forces in Molecules. *Physical review* **56**, 340–343 (1939).

[53] Bartók, A. P. & Csányi, G. Gaussian approximation potentials: A brief tutorial introduction. *International Journal of Quantum Chemistry* **115**, 1051–1057 (2015).

[54] Macêdo, I. & Castro, R. Learning divergence-free and curl-free vector fields with matrix-valued kernels. Tech. Rep. A679/2010, Instituto Nacional de Matematica Pura e Aplicada (2008).

[55] Rupp, M., Tkatchenko, A., Müller, K. R. & von Lilienfeld, O. A. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Physical Review Letters* **108**, 058301–5 (2012).

[56] Rupp, M. Machine learning for quantum mechanics in a nutshell. *International Journal of Quantum Chemistry* **115**, 1058–1073 (2015).

[57] Hansen, K. *et al.* Assessment and Validation of Machine Learning Methods for Predicting Molecular Atomization Energies. *Journal of Chemical Theory and Computation* **9**, 3404–3419 (2013).

[58] Chmiela, S. *et al.* Machine learning of accurate energy-conserving molecular force fields. *Science Advances* **3**, e1603015 (2017).

[59] Glielmo, A., Sollich, P. & De Vita, A. Accurate interatomic force fields via machine learning with covariant kernels. *Physical Review B* **95**, 214302 (2017).

[60] Zeni, C. *et al.* Building machine learning force fields for nanoclusters. *The Journal of Chemical Physics* **148**, 241739 (2018).

[61] Deringer, V. L. & Csányi, G. Machine learning based interatomic potential for amorphous carbon. *Physical Review B* **95**, 094203 (2017).

[62] Huo, H. & Rupp, M. Unified representation of molecules and crystals for machine learning (2017). `arXiv:1704.06439`.

[63] Bartók, A. P., Gillan, M. J., Manby, F. R. & Csányi, G. Machine-learning approach for one- and two-body corrections to density functional theory: Applications to molecular and condensed water. *Physical Review B* **88**, 054104–12 (2013).

[64] Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of Chemical Physics* **134**, 074106–14 (2011).

[65] Haasdonk, B. & Burkhardt, H. Invariant kernel functions for pattern analysis and machine learning. *Machine Learning* **68**, 35–61 (2007).

[66] Haussler, D. Convolution kernels on discrete structures. *Technical Report, UCS-CRL-99-10, University of California at Santa Cruz* (1999).

[67] Hornik, K. Some new results on neural network approximation. *Neural networks* **6**, 1069–1072 (1993).

[68] Szlachta, W. J., Bartók, A. P. & Csányi, G. Accuracy and transferability of Gaussian approximation potential models for tungsten. *Physical Review B* **90**, 104108–6 (2014).

[69] Rowe, P., Csányi, G., Alfè, D. & Michaelides, A. Development of a machine learning potential for graphene. *Physical Review B* **97**, 054303 (2018).

[70] Anderson, T. W. The Non-Central Wishart Distribution and Certain Problems of Multivariate Statistics. *The Annals of Mathematical Statistics* **17**, 409–431 (1946).

[71] James, A. T. A generating function for averages over the orthogonal group. *Proceedings of the Royal Society A* **229**, 367–375 (1955).

[72] Faber, F. A., Christensen, A. S., Huang, B. & von Lilienfeld, O. A. Alchemical and structural distribution based representation for universal quantum machine learning. *The Journal of Chemical Physics* **148**, 241717–13 (2018).

[73] Huang, B. & von Lilienfeld, O. A. Communication: Understanding molecular representations in machine learning: The role of uniqueness and target similarity. *The Journal of Chemical Physics* **145**, 161102–7 (2016).

[74] von Lilienfeld, O. A., Ramakrishnan, R., Rupp, M. & Knoll, A. Fourier series of atomic radial distribution functions: A molecular fingerprint for machine learning models of quantum chemical properties. *International Journal of Quantum Chemistry* **115**, 1084–1093 (2015).

[75] Mendelev, M. I. *et al.* Development of new interatomic potentials appropriate for crystalline and liquid iron. *Philosophical Magazine* **83**, 3977–3994 (2003).

[76] Csányi, G., Albaret, T., Payne, M. C. & De Vita, A. "Learn on the Fly": A Hybrid Classical and Quantum-Mechanical Molecular Dynamics Simulation. *Physical Review Letters* **93**, 175503–4 (2004).

[77] Botu, V. & Ramprasad, R. Adaptive machine learning framework to accelerate ab initiomolecular dynamics. *International Journal of Quantum Chemistry* **115**, 1074–1083 (2014).

[78] Caccin, M., Li, Z., Kermode, J. R. & De Vita, A. A framework for machine-learning-augmented multiscale atomistic simulations on parallel supercomputers. *International Journal of Quantum Chemistry* **115**, 1129–1139 (2015).

[79] Grisafi, A., Wilkins, D. M., Csányi, G. & Ceriotti, M. Symmetry-adapted machine-learning for tensorial properties of atomistic systems. *Physical Review Letters* **120**, 036002 (2018).

[80] Bereau, T., DiStasio, R. A., Tkatchenko, A. & von Lilienfeld, O. A. Non-covalent interactions across organic and biological subsets of chemical space: Physics-based potentials parametrized from machine learning. *Journal of Chemical Physics* **148**, 241706 (2018).

[81] Mehta, M. L. *Random Matrices; 3rd ed.* Pure and applied mathematics series (Elsevier, San Diego, CA, 2004).

[82] Aubert, S. & Lam, C. S. Invariant integration over the unitary group. *Journal of Mathematical Physics* **44**, 6112–21 (2003).

[83] Mercer, J. Functions of positive and negative type, and their connection with the theory of integral equations. *Proceedings of the Royal Society A* **83**, 69–70 (1909).

[84] Fuselier Jr, E. J. *Refined error estimates for matrix-valued radial basis functions*. Ph.D. thesis, Texas A&M University.

[85] Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Physical Review Letters* **77**, 3865–3868 (1996).

[86] Bianchini, F., Kermode, J. R. & De Vita, A. Modelling defects in Ni–Al with EAM and DFT calculations. *Modelling and Simulation in Materials Science and Engineering* **24**, 045012 (2016).

[87] Bianchini, F., Glielmo, A., Kermode, J. R. & De Vita, A. Enabling qm-accurate simulation of dislocation motion in $\gamma-$Ni and $\alpha-$Fe using a hybrid multiscale approach. *Physical Review Materials* **3**, 043605 (2019).

[88] Bianchini, F. *Mechanical Properties of Nickel-based Superalloys A Multiscale Atomistic Investigation.* Ph.D. thesis (2016).

[89] von Pezold, J., Lymperakis, L. & Neugebeauer, J. Hydrogen-enhanced local plasticity at dilute bulk H concentrations: The role of H-H interactions and the formation of local hydrides. *Acta Materialia* **59**, 2969–2980 (2011).

[90] Elstner, M. *et al.* Self-consistent-charge density-functional tight-binding method for simulations of complex materials properties. *Physical Review B* **58**, 7260–7268 (1998).

[91] Jones, A. & Leimkuhler, B. Adaptive stochastic methods for sampling driven molecular systems. *The Journal of Chemical Physics* **135**, 084125–12 (2011).

[92] Hansen, K. *et al.* Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *The Journal of Physical Chemistry Letters* **6**, 2326–2331 (2015).

[93] Shapeev, A. V. Moment Tensor Potentials: A Class of Systematically Improvable Interatomic Potentials. *Multiscale Modeling & Simulation* **14**, 1153–1173 (2016).

[94] Wolpert, D. H. The lack of a priori distinctions between learning algorithms. *Neural Computation* **8**, 1341–1390 (1996).

[95] Jefferys, W. H. & Berger, J. O. Ockham's Razor and Bayesian Analysis. *American Scientist* **80**, 64–72 (1992).

[96] Rasmussen, C. E. & Ghahramani, Z. Occam's razor. In *Advances in Neural Information Processing Systems 13*, 294–300 (2001).

[97] Ghahramani, Z. Probabilistic machine learning and artificial intelligence. *Nature* **521**, 452–459 (2015).

[98] Jacobs, R. A., Jordan, M. I., Nowlan, S. J. & Hinton, G. E. Adaptive Mixtures of Local Experts. *Neural computation* **3**, 79–87 (1991).

[99] Rasmussen, C. E. & Ghahramani, Z. Infinite mixtures of Gaussian process experts. In *Advances in Neural Information Processing Systems 14*, 881–888 (2002).

[100] Mavračić, J., Mocanu, F. C., Deringer, V. L., Csányi, G. & Elliott, S. R. Similarity Between Amorphous and Crystalline Phases: The Case of TiO2. *The Journal of Physical Chemistry Letters* **9**, 2985–2990 (2018).

[101] De, S., Musil, F., Ingram, T., Baldauf, C. & Ceriotti, M. Mapping and classifying molecules from a high-throughput structural database. *Journal of Cheminformatics* 1–14 (2017).

[102] Breiman, L. Bagging predictors. *Machine Learning* **24**, 123–140 (1996).

[103] Sagi, O. & Rokach, L. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **8**, e1249–18 (2018).

[104] Sewell, M. Ensemble learning. Tech. Rep. RN/11/02, Department of Computer Science, UCL, London (2008).

[105] Takahashi, A., Seko, A. & Tanaka, I. Linearized machine-learning interatomic potentials for non-magnetic elemental metals: Limitation of pairwise descriptors and trend of predictive power (2017).

[106] Altland, A. & S., B. D. *Condensed Matter Field Theory* (Cambridge University Press, 2010).

[107] Becca, F. & Sorella, S. *Quantum Monte Carlo Approaches for Correlated Systems* (Cambridge University Press, 2017), 1 edn.

[108] Hubbard, J. Electron Correlations in Narrow Energy Bands. *Proceedings of the Royal Society A* **276**, 238–+ (1963).

[109] Essler, F. H. L., Frahm, H., Göhmann, F., Klümper, A. & Korepin, V. E. *The One-Dimensional Hubbard Model* (Cambridge University Press, 2005).

[110] Hubbard, J. & Flowers, B. H. Electron correlations in narrow energy bands iii. an improved solution. *Proceedings of the Royal Society A* **281**, 401–419 (1964).

[111] LeBlanc, J. P. F. *et al.* Solutions of the Two-Dimensional Hubbard Model: Benchmarks and Results from a Wide Range of Numerical Algorithms. *Physical Review X* **5**, 517–28 (2015).

[112] Rodríguez-Guzmán, R., Jiménez-Hoyos, C. A., Schutski, R. & Scuseria, G. E. Multireference symmetry-projected variational approaches for ground and excited states of the one-dimensional Hubbard model. *Physical Review B* **87**, 96–14 (2013).

[113] Deng, Y., Kozik, E. & Prokof'ev, N. V. Emergent BCS regime of the two-dimensional fermionic Hubbard model: Ground-state phase diagram. *Europhysics Letters* **110**, 57001 (2015).

[114] Sakurai, J. J. & Napolitano, J. *Modern Quantum Mechanics* (Cambridge University Press, 2017).

[115] Hastings, W. K. Monte-Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97 (1970).

[116] Neuscamman, E., Changlani, H., Kinder, J. & Chan, G. K. Nonstochastic algorithms for Jastrow-Slater and correlator product state wave functions. *Physical Review B* **84**, 205132–9 (2011).

[117] Toulouse, J. & Umrigar, C. J. Full optimization of Jastrow–Slater wave functions with application to the first-row atoms and homonuclear diatomic molecules. *The Journal of Chemical Physics* **128**, 174101–15 (2008).

[118] Casula, M. & Sorella, S. Geminal wave functions with Jastrow correlation: A first application to atoms. *The Journal of Chemical Physics* **119**, 6500–6511 (2003).

[119] Sun, Q. *et al.* PySCF: the Python-based simulations of chemistry framework. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **8**, e1340–15 (2018).

[120] White, S. R. Density matrix formulation for quantum renormalization groups. *Physical Review Letters* **69**, 2863–2866 (1992).

[121] Schollwöck, U. The density-matrix renormalization group: a short introduction. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **369**, 2643–2661 (2011).

[122] Olivares-Amaya, R. *et al.* The ab-initio density matrix renormalization group in practice. *The Journal of Chemical Physics* **142**, 034102–14 (2015).

[123] Ruder, S. An overview of gradient descent optimization algorithms (2016). `arXiv:1609.04747`.

[124] Sorella, S. Green Function Monte Carlo with Stochastic Reconfiguration. *Physical Review Letters* **80**, 4558–4561 (1998).

[125] Lawrence, N., Seeger, M. & Herbrich, R. Fast sparse Gaussian process methods: The informative vector machine. In *Advances in Neural Information Processing Systems 15*, 625–632 (2002).

[126] Tipping, M. E. The relevance vector machine. In *Advances in Neural Information Processing Systems 12*, 652–658 (2000).

[127] Snelson, E. & Ghahramani, Z. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*, 1257–1264 (2006).

[128] Boloix-Tortosa, R., Murillo-Fuentes, J. J., Payan-Somet, F. J. & Perez-Cruz, F. Complex Gaussian Processes for Regression. *IEEE Transactions on Neural Networks and Learning Systems* **29**, 5499–5511 (2018).

[129] Larsen, A. H. *et al.* The atomic simulation environment—a Python library for working with atoms. *Journal of Physics-Condensed Matter* **29**, 273002 (2017).

[130] Zeni, C., Fekete, Á. & Glielmo, A. MFF: a Python package for building nonparametric force fields from machine learning (2018). Code: `https://github.com/kcl-tscm/mff`; Documentation: `https://mff.readthedocs.io/en/latest/`; DOI: `https://doi.org/10.5281/zenodo.1475959`.

[131] Tipping, M. E. Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research* **1**, 211–244 (2001).

[132] Tipping, M. E. & Faul, A. C. Fast marginal likelihood maximisation for sparse Bayesian models. In *International Conference on Artificial Intelligence and Statistics 9*, 3–6 (2003).

[133] Mahoney, M. W. & Drineas, P. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences* **106**, 697–702 (2009).

[134] Titsias, M. K. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics 12*, 567–574 (2009).

[135] Hensman, J., Fusi, N. & Lawrence, N. D. Gaussian Processes for Big Data. In *Conference on Uncertainty in Artificial Intelligence 29*, 282–290 (2013).

[136] Darmawan, A. S., Nomura, Y., Yamaji, Y. & Imada, M. Stripe and superconducting order competing in the Hubbard model on a square lattice studied by a combined variational Monte Carlo and tensor network method. *Physical Review B* **98**, 205132 (2018).

[137] Berger, J. O. *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics (Springer Science & Business Media, New York, NY, 2013).

[138] Abramowitz, M. & Stegun, I. A. *Handbook of mathematical functions with formulas, graphs, and mathematical tables* (National Bureau of Standards: Applied Mathematics Series, 1972).